DII.COE3.0.DRAFT.ALL.PM

Defense Information Infrastructure (DII)

Common Operating Environment (COE)

Programmer's Manual

Version 3.0

31 October 1996

Prepared for:

Defense Information Systems Agency

Table of Contents

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

List of Figures

1. Introduction

1.1 Purpose

The *Defense Information Infrastructure (DII) Common Operating Environment (COE) Programmer's Manual* is a broad-based description of the DII COE to effectively communicate how the available services are employed by the developers of mission support applications. This manual has been created, in conjunction with the *DII COE Programmer's Reference Manual*, to provide COE users and programmers with a manual for system and software development within the DII COE environment.

This manual provides an overview of the COE design and development concepts, the requirements and standards that apply in this environment, describes the services currently available through the COE, and a general "how to use" discussion for these services.

The Defense Information Systems Agency (DISA) will review and update this document as required to remain current with the evolution of the DII COE. This document supersedes Version 2.0, all earlier draft versions, presentations, or working group notes. Please direct any comments or questions regarding the *DII COE Programmer's Manual* to:

Point of Contact (POC)	Cmdr. Charles B. Cameron Chief, DII COE Engineering Division
Address	DISA/JIEO/JEXF-OSF 45335 Vintage Park Plaza Sterling, VA 20166-6701
Telephone	(703) 735-8825
Fax	(703) 735-8761
Internet Address	cameronc@ncr.disa.mil

1.2 Background

The national military strategy dictates that United States (US) forces be able to project power from Continental US (CONUS) bases, sanctuaries, and in-theater locations to an area of conflict. In addition, US forces must support humanitarian missions, often on short notice, anywhere in the world. These military missions make the Services increasingly reliant on long-distance communications and information services of all kinds. The DII is intended to provide the Military Services (and all other Department of Defense (DoD) elements) with information management capabilities to meet these and other DoD missions. The DII COE is the mechanism used to build and integrate the systems and services of the DII.

The purpose and vision of the DII are detailed in the 28 July 1995, DII Master Plan, Version 3.0, which summarizes the primary guidance to the DII integration effort. The DII is defined as a

seamless web of communication networks, computers, databases, applications, and the associated people and facilities that meet the information processing and transport needs of DoD users in peace and all crisis, conflict, humanitarian support, and wartime roles. It includes: (1) the facilities to transmit, store, process, and display voice, data, and images; (2) the data, such as video programming, databases, and other media; (3) the applications and software; (4) the network standards and protocols; and (5) the resources to design, develop, construct, manage, and operate the DII.

The DII is a departmental asset. It is the sum of all parts that constitute the information management assets owned by each of the components, including the Office of the Secretary of Defense (OSD), the Joint Chiefs of Staff, the individual Services, Agencies, and others. The Principal Staff Assistants (PSAs) (including the Joint Staff), the Commanders in Chief (CINC), Services, and Agencies (C/S/As), and DISA share the responsibility for policies, budgets, installations, and operations of the DII. PSAs are responsible for planning and funding functional applications while DISA is responsible for the implementation of the support infrastructure. These applications depend on the shared infrastructure provided by the DII COE.

1.3 Scope

This document describes the concepts, models and architecture of the DII COE, its major components, and the resources provided for COE development activities. It also describes the requirements for building and integrating software components on top of the DII COE. This document provides implementation concepts which describe, from the perspective of DII development as follows:

- DII COE concept, model, and principles,
- major components of the DII COE,
- resources provided for COE development activities,
- requirements for COE software compliance and registration, and
- DII COE runtime environment.

1.4 Document Structure

This *DII COE Programmer's Manual* is divided into seven sections. Each section is devoted to a topic of primary importance in DII COE development activity.

Section 1 discusses the purpose, background, and scope of this document. It also contains an overview of the DII.

Section 2 describes the concepts and requirements for the DII COE to support DoD services and mission applications, the COE design and development considerations, and an introduction to the *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification* (referred to as the *DII COE I&RTS*).

Section 3 describes the functionality provided by the DII COE Kernel and introduces the COE environment. It includes an introduction to the Applications Program Interfaces (API) tools available to assist the COE application developers.

Section 4 describes the developer's toolkit and application program interfaces (API) and identifies the various elements delivered in the DII COE support services.

Section 5 describes the support infrastructure services consisting of management services that a developer may use in programming various mission and component software.

Section 6 describes support applications that are available in creating DII COE segments and identifies the major applications to essential functions including the desktop office automation.

Section 7 provides basic information resources in support of the users of the DII COE including user information for the extension of the COE, security, and on-line services.

Appendices There are five appendices (A through D) which contain the following:

A: Acronym List
B: Glossary
C: References
D: On-line Information Resources
E: List of TAFIM Adopted Standards

This page intentionally left blank.

2. DII COE Concept Overview

The DII effectively integrates people and organizations, data and information, and work processes and computing services across the DoD. The DII COE consists of architecture, standards, and reusable software components. The DII COE is a "plug and play" open architecture designed around a client/server model. The DII COE concept is best described as an architecture that is fully compliant with the *Department of Defense Technical Architecture Framework for Information Management* (TAFIM) and meets the requirements of the DoD Joint Technical Architecture (JTA).

The DII COE concept provides a distributed application infrastructure that promotes DoD-wide interoperability, portability, and scalability. Its objective is to provide access to data, independent of location, in a reliable, cost-efficient manner. Refer to Chapter 1 of the *DII COE I&RTS*, Version 2.0 dated October 23, 1995, for details about what the COE encompasses.

2.1 The DII COE Foundation

The COE is an Open System Environment (OSE) that provides the 'foundation' for building an open system and provides a practical update mechanism for operational sites. Figure 2-1 shows how the DII COE serves as a foundation for building multiple systems. The diagram shows two types of reusable software: the operating system, and the COE components. Section 2 of the *DII COE I&RTS* describes the COE components in more detail. It is sufficient to note that access to these components is through Application Program Interfaces (APIs) and they form the architectural backbone of the target system. Each system built upon the COE foundation uses the same set of APIs to access common COE components, the same approach to integration, and the same set of tools for enforcing COE principles. Precisely the same COE software components are used for common functions, such as communications interfaces and data flow management.

The DII COE is an open architecture that is not tied to a specific hardware platform. It uses POSIX-compliant operating systems and industry standards such as X-Window and Motif. The present COE taxonomy and architectural design requirements are detailed in the *Architectural Design Document for the Defense Information Infrastructure (DII) Common Operating Environment (COE*), dated January 1996.

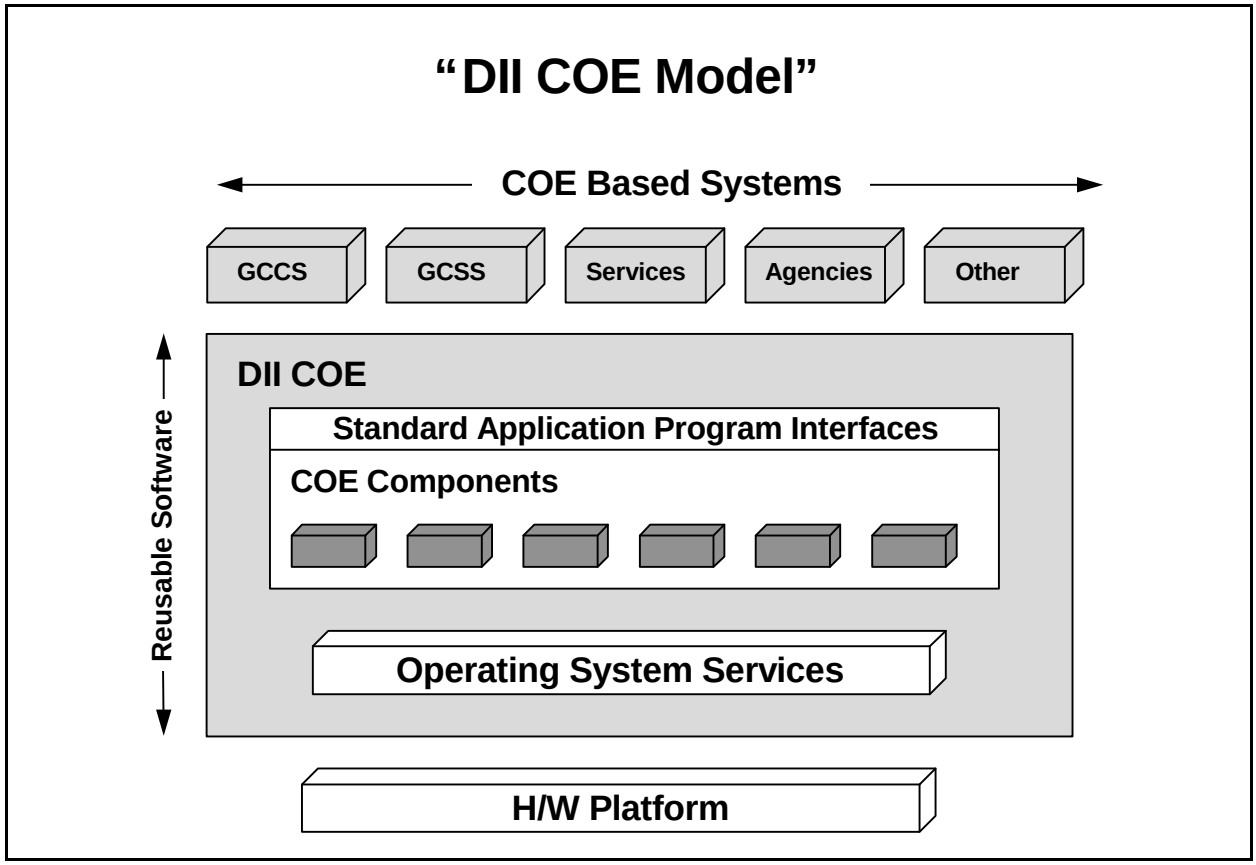

Figure 2-1. DII COE and COE Based Systems

The DII COE architecture is divided into two major areas: Platform Services and Common Support Applications. Platform Services are those types of services that support the flow of information, while Common Support Applications are critical to interoperability but do not directly support the flow of information.

Platform Services include the following functions:

- Management Services
- Security Services
- Communications Services
- Distributed Computing
- Data Management Services
- Presentation Services

Common Support Applications include the following functions:

- Office Automation
- Mapping, Charting, Geodesy and Imaging (MCG&I) Service
- Message Processing Service

- Alert Services
- Correlation Service
- Situation Display Service

The COE is also an evolutionary acquisition and implementation strategy. It emphasizes incremental development and fielding to reduce the time required to put new functionality into the hands of the warrior, while maintaining software quality and minimizing program risk and cost.

2.2 COE Concept

In COE-based systems all software, except the operating system and basic windowing software, is packaged in self-contained units called 'segments' Segments are the most basic building blocks and are defined in terms of the functions they perform. They may contain one or more Computer Software Configuration Items (CSCI). Reusable segments that are part of the COE are known as COE components or simply 'components'. The principles governing how segments are loaded, removed, or interact with one another are the same for all segments. Selecting software modules that fulfill the COE component requirements, and extending them to meet other problem domains is an ongoing task. The COE preserves backward compatibility so that mission applications are not abandoned just because there is an update of the COE.

The DII COE is structured as a "plug and play" architecture. Migration of existing legacy systems to the COE is conceptually straightforward, but may require considerable effort due to the requirement to switch to a different set of building blocks. That is, the effort may not be in conforming to a new architectural concept but in modifying code to use a common set of APIs. The 'plug and play' concept clearly conveys the goal and the simplicity that most segment developers will encounter. The *DII COE I&RTS* Chapter 2 clarifies the basic understanding of the segments and the distinction between interoperability and integration.

2.3 Model & Principles

This subsection introduces the DII COE Model and Structure of the development environment. It discusses the basic principles and structure of the DII COE. Additional details about the DII COE tools and APIs can be found in the *DII COE Programmer's Reference Manual*.

Figure 2-2 illustrates the relationship between the COE, component segments, and mission application segments. The COE encompasses Government-Off-The-Shelf (GOTS) software, Commercial-Off-The-Shelf (COTS) software, the operating system, windowing software, tools and APIs. Refer to the *User Interface Specifications for the Defense Information Infrastructure (DII)* (*DII COE Style Guide*),Version 2.0 dated 1 April 1996, the *DII COE I&RTS*, and the *Architecture Design Document for the DII COE* for more details about COE model and principles. Figure 2-2 is a generic diagram intended only to show relationships and relative position of the services, APIs, and the mission applications. The labeled boxes show services can be useful for the Command, Control, Communications, Computers, and Intelligence (C4I), logistics, and mission application domains while the principles remain unchanged. The diagram

remains applicable even when service boxes are replaced with examples of another problem domain.

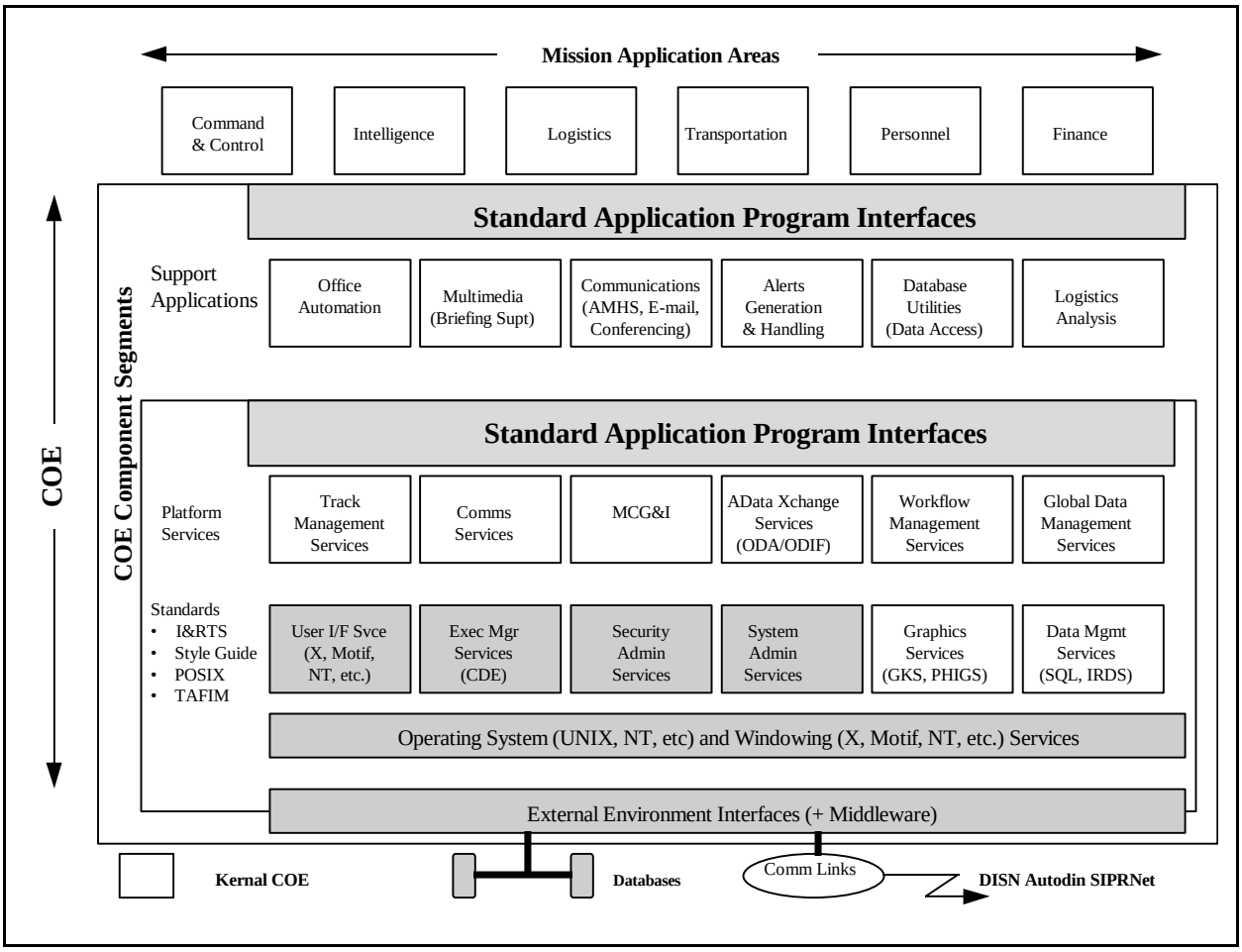

Figure 2-2. DII COE Model

The COE is a collection of building blocks where the operating system, windowing environment, and COE core services form the COE backbone (called the kernel) that other COE component segments "plug" into. Some of the COE components are required, such as the kernel components, while other components are optional depending upon the system being built. Selection of the specific software modules which comprise the COE determines which problem domain(s) can be addressed by the particular COE installation. Chapter 2 of the *DII COE I&RTS* details the four functions for selecting software components.

The DII COE APIs define how segments may interconnect. Figure 2-2 also implies that APIs are the only avenue for accessing support application and platform services provided by the COE. This is true for all COE software, including COTS software. However, the COE does not create an additional layer on top of the COTS software. These components may be accessed directly using vendor-supplied APIs for these commercial products as long as such usage does not

circumvent the COE model (i.e., conflict with the COE runtime environment). For example, the COE supports POSIX-compliant operating systems. Some vendors provide non-POSIX compliant extensions to the operating system services. Use of such extensions, even when readily available through vendor supplied APIs, is not allowed since such usage violates the COE architecture and inhibits interoperation, portability, and reuse.

Physical databases (e.g., MIDS, IDB, JOPES database, etc.) are not considered part of the COE although the software, such as the relational database management system (RDBMS) which accesses and manages the data, is part of the COE. Within this conceptual model, the RDBMS functions as the COE's disk controller and disk drives. The application's databases, which are not part of the COE, can be equated to directories or partitions on the drives accessed through the RDBMS disk controller. Data objects belonging to each database can be considered as files within those directories.

The precise configuration of COTS products used in the COE is under strict configuration control. This is necessary because configurable items, such as the amount of shared memory or swap space, must be known and carefully controlled in order for other components in the COE to operate properly. For this reason, COTS products are assigned a DII COE version number in addition to the vendor-supplied version number so as to be able to track and manage configuration changes.

A fundamental principle throughout the COE is that segments are not allowed to directly modify any resource "owned" by another segment. This includes files, directories, modifications to the operating system, and modification to windowing environment resources. Instead, the COE provides tools through which a segment can request extensions to its base environment. The importance of this principle cannot be overemphasized because environmental interactions between software components are a primary reason for integration difficulties. By providing software tools which arbitrate requests to extend the environment, integration can be largely automated and potential problem areas can be identified.

2.4 Design and Development Considerations

This subsection describes the essential elements for the design and development of applications using the DII COE tools and APIs. It is not intended to be a comprehensive guide for programming. Rather, it provides the concepts and considerations in the design of client/server-based distributed applications for DII. For more comprehensive design and development for life-cycle maintenance, refer to the Software Engineering Institute (SEI) Capability Maturity Model (CMM).

2.4.1 System Concepts for Programming

Segments provide functionality that is easily added to or removed from the target system in small manageable units. Segments are defined in terms of functions that are meaningful to operators, not in terms of internal software structure. Structuring the software into segments is a powerful

concept that allows considerable flexibility for configuring the system to meet specific mission needs or to minimize hardware requirements for an operational site.

Integration is not possible without strict standards that describe how to properly build components to add to the system. The requirements for DII COE compliant segments are defined in the *DII COE I&RTS* and the *DII COE Style Guide*. The DII COE provides automated tools to measure compliance and to pinpoint problem areas. A useful side effect of the tools and procedures is that software integration is largely an automated process, thus significantly reducing development time while automatically detecting potential integration and runtime problem areas.

More precisely, to a developer the DII COE is:

- An Architecture
- A Runtime Environment
- Software Development
- Tools & APIs

2.4.2 Building for Integration

Integrated applications are the main focus of the DII mission because of their importance to the DoD efforts in applications interoperability, portability, scalability, and component reuse. Integration of mission applications is paramount to system operations and the user community within the DII. However, system's application components within a platform require a high level of integration to assure component independence of software versions, enhancements and upgrades.

2.4.2.1 Rightsizing of Applications

"Rightsizing" requires the optimum division and allocation of computing tasks between client and server for distributed applications. Rightsizing also takes into consideration the requirements for custom-designed mission applications and the integration of DII COE components. Consideration of the network service requirements is also essential in providing a context for a distributed application design. The division of computing tasks is a design issue in mission applications and may affect the relative stability of the COE components used.

A well-planned integrated application is network infrastructure dependent and provides an environment that may grow or shrink dynamically or progressively, from local to regional and global interactions, as the DoD mission requirements change. The progression from simple (local) to universal (global) use increases the complexity of applications and system deployment, operations, and support incrementally. However, the basic distributed applications, as designed, should not change in the process.

2.4.2.2 Application/Data Separation

Achieving a high level of application portability and component reuse requires development methods and tools that guarantee a level of independence. The separation of application processing from the semantics of data is a primary consideration to develop programs that are independent of the vendor-specific database or data warehouse in use. Similarly the DII COE infrastructure (as defined by TAFIM and related OSE standards) requires the independence of application components from operating system services. Separation of those services that are native to the computer in use, versus standards-defined POSIX services and interfaces, is the key. This separation assures the means to create portable and reusable applications components and common functional modules (or subroutine calls) that interact with mission applications to provide the highest degree of cross-application integration.

2.4.2.3 Design for Reuse and Porting

The DII COE application development services support the reuse and portability of mission applications. These services use concepts, methods and infrastructure defined by the TAFIM and related OSE standards. The COE provides the means to create portable and reusable applications components (and common functional modules) that interact with other applications residing in different computers regardless of the platform.

The most important aspect of developing mission essential portable applications is to scope the tasks and clearly define the objectives of the client and the server. A distributed application that uses OSE application services must use APIs. APIs are primarily standards-based and provide the appropriate programming language bindings (i.e., Ada and C).

Specialized or mission critical distributed application capability requires an in-depth understanding of the business process and operational requirements. But foremost, the designer must understand the functional limitations of the environment that is used or created.

Understanding the methods and automating the business/operations processes are not sufficient to develop an optimum distributed processing environment. Sometimes, a redesign of the business process is necessary to maximize the benefit of distributed computing. Another aspect of designing and developing a distributed application is understanding the current capabilities of the network services and the planned future enhancements. This is important because applications need to scale and transition to future technologies and new mission realities without excessive constraints. Some considerations and elements are as follows:

- Understanding the mission and the processes required,
- Network responsiveness to the enterprise needs,
- Seamless distributed processing across the network,
- “Reachability” -- local, area, regional, national, international.

2.4.3 Compliance

The COE compliance requirements ensure "plug and play" integration of segments into the COE runtime environment, graphical user interface (GUI) consistency, architectural compatibility, and software quality. Levels of compliance for each of these categories are measured objectively through a set of checklists. These levels allow legacy systems to be migrated into the COE in stages leading up to Full COE Compliance. In addition, the COE provides a suite of tools for validating COE conformance. For more details on compliance, refer to the *DII COE I&RTS*.

2.4.4 Environment Stability

In designing and developing distributed applications, several aspects of the systems environment must be considered to assure a maximum life-cycle for the data (information), the applications (software) and the expertise (human). The issues listed below must be considered when building a durable and robust environment for the design of portable mission applications. At a high level, these issues are:

- Platform - Hardware/Software -- Operating Environment
- Standards for OSE Distributed Computing Requirements
- Vendors/Suppliers' Products -- Availability & Scalability
- Enterprise Environment -- Constraints
- Human Factors Engineering and Man-Machine Interface (GUI)
- Portability and Interoperability
- Mobility and Mission Considerations

2.4.5 Division of Computing Tasks

To build distributed applications in a client/server environment, it is necessary to determine the division of the computing tasks required and where these tasks will reside (i.e., the client or the server). The fundamental concept is to analyze the tasks required for the application and logically unbundle the distinct processes that comprise the entire application. For example, a traditional application may have the following components: user interface, application logic, data manipulation or management, data repository (disk device handler), and various data output media. The application logic may be further subdivided into data validation functions and computational logic. These functions may reside individually in separate processors tied together by a communications facility. The communication facility may be a Local Area Network (LAN), a Wide Area Network (WAN), or both. The communications facility is provided by a suite of protocols that can span across network nodes that reside within LANs and WANs.

Figure 2-3 shows the division of tasks between clients (requesters) and servers (responders) or between various servers. This division can be viewed as sequences of requests and responses. The client task can exist independently on its own (self-sufficient), but may require the server to complete a distributed task. The various models of task division may cross other application processes and be shared by cooperating systems. The paradigm of using computers that serve both client and server roles gives way to developing specialized applications that specifically

optimize the use of the processor's capabilities and resources. Therefore, the demarcation line of the division of labor (tasks) may vary between different application components, but must remain consistent within a specific application distributed across the network.

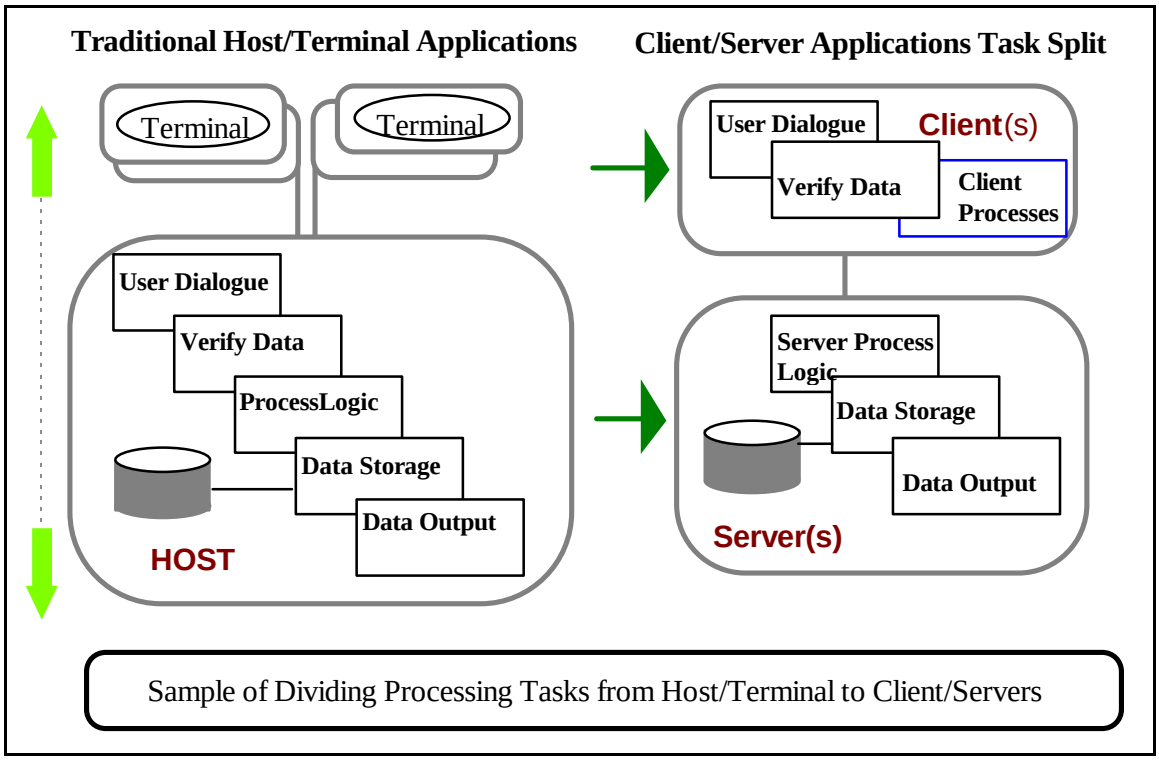

Figure 2-3. Division of Client/Server Tasks

2.4.6 OSE Enabled COE Applications

The TAFIM is based on OSE Guide (IEEE P1003.0) and is the foundation of the DII COE. OSE application services and protocols enable various mission and business application processes to interface over diverse networks while residing on heterogeneous platforms. The advantage of OSE and DII COE concepts is that the applications themselves provide specific system-level services that may be used selectively with or without operator intervention. The user may be interactive via a user-interface facility (i.e., GUI), another specialized mission, application, or an operational application that needs the OSE services.

For instance, the use of X.400 (Message Handling System) provides the transport mechanism for compound documents or multimedia binary objects using a store-and-forward mechanism. This transport mechanism works like the postal service to relieve the sender of the need to monitor the delivery of the message and its objects to its destination. Similarly, a mission application may choose to send an electronic message (E-mail) automatically in the form of a transaction notification. This is accomplished by inserting a formatted or free-form message directed to another process or entity with assurance of delivery and arrival status of the messages sent.

Mission applications may use some or all of the service facilities of the COE suite. Besides the use of typical E-mail transport, COE application facilities include: network utility-type services for Distributed Directory, File Transfer, Remote Data Access, Transaction Processing, and specialized application layer services such as Remote Procedure Calls (RPC), Open Software Foundation/Distributed Computing Environment (OSF/DCE), Manufacturing Message Services or Information Storage and Retrieval that provide COE enabled applications across DII.

2.4.7 Distributed COE Applications

The COE application services were intended to be used in OSE as generalized utilities to serve the needs of various mission and enterprise applications. The concept of using OSE applications as generic application utilities has matured and can be used by programmers today by way of functional calls using APIs.

An overwhelming majority of programmer-written applications require only basic communication services: 1) open connection, 2) send and receive messages 3) close connection, and 4) handle errors. In a connection-oriented mode the outcome of any communication service request is always acknowledged.

All network services and communications-based application protocols provide utilities and interface services that can be used to support the development of distributed mission applications. The application developer can write applications that take advantage of these services thus enhancing the customer's application development environment to meet their specific mission requirements. By using these generic network application services, programmer-written applications can provide a strategic advantage to DoD Agencies/Services Information Resource Managers (IRM) in overcoming the backlog of software development.

The distribution of computing tasks as "peer-level" computing services allows software on multiple, independent, heterogeneous processors to cooperate in providing effective delivery of application messages or services. COE provides the foundation services for building the DII and the means to create OSE compliant applications. DII COE provides the ability to work cooperatively across multiple, heterogeneous systems and platforms to accomplish the DoD mission.

2.5 Integration and Runtime Specification

The *DII COE I&RTS* delineates technical requirements for using the DII COE to build and integrate systems. It provides additional details on many topics covered in this manual and in the Programmers Reference Manual. The *DII COE I&RTS* provides detailed implementation details that describe, from a software development perspective, the following:

- COE approach to software reuse,
- COE runtime execution environment,
- definition and requirements for achieving COE compliance,
- process for automated software integration, and

- process for electronically submitting/retrieving software components to/from the COE software repository.

2.5.1 Runtime Environment

The runtime environment encompasses all aspects of the software configuration for the COE. All software and data, excepting low level components of the bootstrap COE, are packaged as segments. Segments are constructed to keep related configuration items together so that functionality may be easily included or excluded. There are nine segment types and attributes corresponding to the different types of components that may be added to a system: COTS, Data, Database, Account Group, Software, Patch, Aggregate Attribute, COE Component Attribute, and Segment Dependencies.

Segment installation is accomplished in a disciplined way through instructions contained in files provided with each segment. These files are called *segment descriptor files* and are contained in a special subdirectory. Installation tools process the segment descriptor files to create a carefully controlled approach to adding or deleting segments to or from the system. The format and contents of the segment descriptor files are the central focus of the runtime environment. Developers are required to adhere to certain procedures to ensure that segments can be installed and removed correctly, and that the installation or removal of segments do not adversely impact another segment.

2.5.1.1 Segment Types

Segment types and attributes are the cornerstone of the COE approach. Developers have considerable freedom in building segments. The DII COE segment types as defined in the *DII COE I&RTS* are:

- COTS Segments
- Data Segments
- Database Segment
- Account Group Segments
- Software Segments
- Patch Segments
- Aggregate Attribute
- COE Component Attribute
- Segment Dependencies

2.5.1.2 Segment Prefixes and Reserved Symbols

Each segment is assigned a unique subdirectory called the *segment's assigned directory*. The assigned directory serves to uniquely identify each segment, but it is too cumbersome for use in naming public symbols. Therefore, each segment is also assigned a 1-6 character alphanumeric string called the *segment prefix*. The segment prefix is used for naming environment variables and

for public APIs and public libraries, where naming conflicts with other segments must be avoided. All segments shall preface their environment variables with the segment's assigned prefix.

The segment prefix is also used to uniquely name executables. All COE component segments shall use the segment prefix to name executables, and it is strongly recommended that all segments follow the same convention. This approach simplifies the task of determining the files that go with each segment and reduces the probability of naming conflicts.

2.5.1.3 Disk Directory Layout

A standardized disk directory structure for all segments is required to prevent two segments from overwriting the same file, creating two different files with the same name, or similar issues that cause integration problems. Unfortunately, such problems are often not discovered until the system is operational in the field.

In the COE approach, each segment is assigned its own unique, self contained subdirectory. A segment is not allowed to directly modify any file or resource it does not "own" (i.e., outside its assigned directory). Files outside a segment's directory are called *community files*. COE tools coordinate modification of all community files at installation time, while APIs attach to the segments that are used at runtime.

The three main types of subdirectories are:

- Segment Subdirectories
- Users Subdirectories
- Developer Subdirectories

2.5.2 Variants

The exact configuration of segments installed on a workstation is called a *variant*. A variant is simply a collection of segments grouped together for installation convenience. The kernel COE is required to be on each workstation or server, but additional segments are dependent upon the functionality desired and how the workstation or server will be used.

2.5.3 System Management

The COE provides templates for 'account group segments' to be used in establishing individual login accounts. Account groups contain template files for defining what COE processes to launch at login time, what functions are to be made available to operators, and preferences such as color selections for window borders. Account groups can also be used to perform a first level division of operators according to how they will use the system. This technique is used in the COE to identify at least five distinct account groups:

- Privileged Operator (e.g., root),
- System Administrator,

- Security Administrator,
- Database Administrator, and
- Non-Privileged Operator.

Additional details on the functions available to each group are in the *I&RTS* document; however, command-line access for privileged operator accounts is expressly prohibited without prior approval from the DISA Chief Engineer. Within an account group, subsets of the available functionality can be created. These subsets are called *profiles*. An operator may participate in multiple account groups with multiple profiles, and can switch from one profile to another without the need to logout and login again. Other account groups may exist for specialized system requirements, such as providing a character-based interface, but all account groups follow the same rules.

2.5.4 Development Process

A powerful feature of the overall development process is the concept of automated integration. This means that automated tools are used to combine and load segments, make environmental modifications requested by segments, make newly loaded segments available to authorized users, and identify places where segments conflict with each other. Traditional system level integration then becomes primarily a task of loading and testing segments. Traditional integration tasks are pushed as far down to the developer level as possible.

Prior to submitting a component to DISA, a developer must:

- package the component as a segment,
- demonstrate COE compliance through tools and checklists,
- test the segment in isolation with the COE,
- provide required segment documentation, and
- demonstrate the segment operating within the COE.

The COE approach is designed to be non-intrusive; it places minimal constraints on how developers build, test, and manage software development. It concentrates on the end product and how it will integrate with the overall system. This approach provides the flexibility to allow developers to conform to their internal development process requirements. However, developers are expected to use proven software engineering practices and design tools to ensure robust products. Developers are free to establish a software development environment that is best suited to their project. The COE requires only that deliveries are packaged as segments, that segments are validated before submission, and the segments are tested in the COE prior to submission. The *DII COE I&RTS* provides detailed lists of requirements and suggestions regarding coding practices, development file structures, and libraries for scripts and files that developers must be familiar with to build a successful DII COE Application.

2.5.5 Database Considerations

Typically, COE-based systems are heavily database oriented. Database considerations are therefore very important in properly architecting and building a system. For more detailed technical information on properly designing databases and database applications see the *DII COE I&RTS*.

2.5.5.1 Database Segmentation Principles

A COE database server is provided by a COTS RDBMS product. It is used in common by multiple applications, and is a services segment and part of the COE. However, different sites need varying combinations of applications and databases. As a result, databases cannot be included in the Database Management System (DBMS) segment. Instead, these component databases are provided in a database segment established by the developer. The applications themselves are in a software segment, also established by the developer, but separate from the database segment. If the data fill for the database contains classified data, that data fill must be in a separate data segment associated with the database segment.

Database Segments

The RDBMS is provided as one or more COTS segments. These segments contain the RDBMS executables, the core database configuration, database administration utilities, RDBMS network executables, and development tools provided by the RDBMS vendor. Databases are provided as database segments. These segments contain the executables and scripts to create a database, and tools to load data into the database.

Database Segmentation Responsibilities

Three groups are involved in the implementation of database segments: DISA, the application developers, and the sites' database administrators. The developers and DISA work together to field databases and associated services for the database administrators (DBAs) to maintain. DISA provides the RDBMS as part of the COE. Developers provide the component databases. Sites manage access and maintain the data. Users interact with the databases through mission applications and may, depending on the application, be responsible for the modification and maintenance of data in the databases.

2.5.5.2 RDBMS Tuning and Customization

The core RDBMS instance is configured and tuned by DISA based on the combined requirements of all developers' databases. This allows the RDBMS Server Segments to be reasonably independent of particular hardware configurations and ignorant of specific application sets.

The final tuning of the RDBMS cannot be accomplished until a complete configuration is built and it has an operational load. Developers should provide information for the tuning process, but should not make their applications dependent on particular tuning parameters. Where a

nonstandard parameter is required for operations, developers must provide that information to DISA so the RDBMS services segment can be modified accordingly.

Developers shall not modify the core RDBMS instance's configuration. Extensions or modifications of that configuration require the specific approval of the DISA DII COE Chief Engineer.

If developers modify any of the executable tools (e.g., add User Exits to Oracle*Forms), then the modified version of the tool does not reside with the core database services, but becomes a part of the application's client segment. This prevents conflicts among different modified versions of a core function. The maintenance of that modified tool also becomes the responsibility of the developers.

2.5.5.3 Database Inter-Segment Dependencies

A key objective of the segmentation approach is to limit the interdependencies among segments. Ideally, database segments should not create data objects in any other schema or own data objects that are dependent on other schemas. However, one purpose in having a Database Server is to limit data redundancy and provide common shared data sets. This means that there will usually be some dependencies among the databases in the federation. For additional information, see the *DII COE I&RTS*.

This page intentionally left blank.

3. DII COE Kernel

The COE *kernel* is the mandatory minimal set of software required on every workstation to provide certain common services no matter which workstation will be used. The kernel COE components are the shaded boxes shown in Figure 2-2. The base of the Kernel is a POSIX-compliant vendor-supplied operating system for a specific COE platform. This operating system includes a windowing user interface modeled after the X-11 Window standard. Additional commercial products and COE segments are also added to provide required functionality.

3.1 Operating System

The operating system performs the basic functions of resource allocation and processing for external devices attached to the workstation. Additionally, the POSIX compliant operating system provides a set of basic functions that provide the following functionality:

Process Control - Supports the creation, execution, termination of processes, and provides for the coordination of and communications between processes. POSIX function calls are available to create a new process or to call an executable module; suspend execution until a specified event occurs or for a specified interval; pipe information between processes; extract and manipulate process identification; and terminate a process.

File Operations - Supports the creation, deletion and processing of files and directories. POSIX function calls are available to create files or directories; move through a directory structure; obtain descriptor information regarding a file or check the status of a file; modify the name, status or attributes related to a file; input from or output to a specified position within a file; terminate input/output operations and purge pending operations; remove an existing file or directory from the file system.

Signal Processing - Supports the activation of processes on a timed basis or in response to external events. POSIX function calls are available to manage groups of signals, schedule or wait for signals, examine unprocessed signals, or detect and process blocked signals.

System Parameters - Supports the overall management of configurable system parameters. POSIX function calls are available to determine or set the input and output data rates, or obtain configurable system parameters including the current system date and time.

Group and User IDs - Supports basic security functions inherent in the operating system. POSIX function calls are available to control use of group IDs and associated privileges, or determine and set group or user IDs.

Additional details can be found in Section 6 of the *DII COE Programmers Reference Manual*.

3.2 Windowing

In addition to a basic operating system, each COE kernel provides a Windowing environment for the Graphical User Interface (GUI). The standard for this interface is the X-11 Window standard. Windowing implementations include Windows@NT, X-Window, and MOTIF. Windowing systems provide the following functionality:

Menus - Allows for the generation of custom menus within the desktop environment and the customization of existing applications in the desktop.

Icons - Allows for the creation of pictorial representations of applications associated with the information which the operating system needs to execute the application.

3.3 Desktop

A graphical user environment is provided for all platforms. For UNIX systems, the Common Desktop Environment (CDE) was jointly developed by IBM Corporation, Hewlett-Packard Company, Novell Inc., and SunSoft Inc. It defines a common set of APIs for a CDE that can be implemented in operating environments that can support X-Window desktop and OSF/Motif. The CDE is implemented for UNIX platforms using a commercial product from TriTeal Enterprises. Platforms running Microsoft Windows@ NT use the desktop that is generic to that system and a suite of office programs. The Desktop helps in bringing every application, network resource, and Internet service into one integrated GUI throughout the systems.

3.3.1 Desktop Tools

The desktop environment provides the following functionality:

Mail Tools - Allows the user to compose, view, and manage electronic mail through the GUI. Allows the inclusion of attachments and communications with other clients through the messaging system.

Calendar Manager - Allows the user to manage, schedule, view appointments, and create a calendar. Meetings and other events can be interactively scheduled with the Mail Tool.

Workspace Manager - Allows the user to organize work into multiple workspaces and navigate between the workspaces.

Editor - Allows the user to create, update, and manipulate text with common functionality including clipboard interaction with other applications.

Terminal Emulator - Allows the user to emulate X-term-like terminals that support terminal emulation of ANSI X3.64-1979 and ISO 6429:1992(E) compliant terminals and to log onto specific host systems.

Calculator - Provides the user with standard calculator functions, using screens and keyboards for inputs and printing.

File Manager - Provides the user with a graphical interface with the file system, including "drag-and-drop" functionality between objects and between cooperating client applications, and a general file type association database.

Print Manager - Provides a simple GUI print job manager to schedule and manage print jobs on any available printer.

Help System - Provides the user with a context-sensitive graphical help system for the desktop.

Style Manager - Allows preferences, such as colors, backdrops, and fonts, to be set interactively for a session using a GUI interface.

ToolTalk - Provides a messaging service that allows communication between applications.

Session Manager - Allows the user to save the current desktop configuration upon logout and restarts the entire session exactly as it was, including applications that were running and the location and size of the windows.

Login Manager - Controls access to the desktop with system password protection and supports password aging and Kerberos.

Application Builder - Provides development tools for constructing applications to run in the desktop environment.

3.3.2 Common Look and Feel

Under the DII concept (refer to the *DII COE Style Guide*), the standardization of the desktop can increase user productivity, reduce training requirements, and increase the efficiency in the development of individual applications. The commonality in "look and feel" is a key element in the usability of an application and is central to the user interaction with and understanding of a system's capabilities.

An application can be viewed as the software available to the user to perform a set of related tasks. This software is visible to the user as a collection of window families, each providing the functionality (in terms of objects and information) needed to perform a particular task. In addition, it is possible for applications to share the services provided by a segment when the applications perform common tasks.

Compliance with the specifications contained in the *DII COE Style Guide* is required in the development of a desktop. All software is expected to comply with the *DII COE Style Guide*

specifications, with deviations occurring only when called for by operational requirements and approved by DISA.

The *DII COE Style Guide* provides guidelines that apply to new development of the GUI in addition to the existing *DoD Human Computer Interface Style Guide* (i.e., Volume 8 of the TAFIM). The *DII COE Style Guide* provides domain level specifications for the 'look and feel' of the software:

- Buttons
- Scrolled windows
- Application Icons
- Text fields
- Command box, selection box and message box
- Frames, labels and panel window
- Font sets
- Color sets

The *DII COE Style Guide* also provides guidance related to user interface internationalization, design of on-line user documentation, and user interface functionality in common support applications. Appendix D and H of the *DII COE I&RTS* map specifications to each of the style-related items included in the COE compliance checklist contained in the *DII COE I&RTS*.

3.4 System Administration

The System Administration function is used to select and configure printers, manage print jobs, and close windows; reboot or shut down the system, mount file systems, and format hard drives; load or install segments; and change machine ID, edit host information, set system time, configure a workstation as either a client or a server, set routing configuration, and configure mail on a workstation. The System Administrator's account group sets the System Administrator's environment to execute the functions listed above.

3.4.1 COE Runtime Tools

The basic set of tools to load and install segments is provided by the DII COE and described in Appendix C of the *DII COE I&RTS*. These tools provide the System Administrator with the following functionality:

Segment Installer - Allows System Administrator to easily perform the following: execute pre-install script, install segments, execute post-install script, execute deinstall script, clone an installation to another workstation with an installed bootstrap COE, install a segment to or from a remote workstation, or update a segment's home environment to reflect the installed location.

Information Extractor - Allows the System Administrator to view segment information reflecting the installed segment baseline and associated information regarding the

installation of new program segments. This tool can be used to display the location and attributes of a named segment, list other segments required to support a specified segment, determine the host name and address of a server for remote installation.

Script Enhancements - Allows for interaction with the System Administrator during the installation process. This tool will prompt the user for textual or Yes/No responses, displays informational and error messages, and provides a special prompt for password entry (response not echoed to screen).

Lists - Allows the System Administrator to view associations between segments and authorized users and/or user/groups. The tool lists all accessible segments on a specified server, lists local segments accessible by a specified profile and user accounts assigned to a profile, lists all profiles or users that can access a specified segment, or lists all profiles or segments assigned to a specific user.

3.4.2 Print Services

Each printer is managed by a designated workstation that handles all print requests directed to that printer. Every COE-based UNIX workstation runs a "printer agent" to communicate between the workstation and a printer server. The printer agent provides the following functionality:

Printer Control - Supports an application in controlling printer resources. It will start a print job and establish context (security classification, identification, page characteristics), advance printing operations to the start of the next page, or signal completion of a print job to initiate the actual printing.

Printer Identification - Supports an application in determining the current printing environment. It provides the name, type (ASCII, Postscript, HPCL), and description of the current default printer.

Transfer Data - Provides a method for an application to transfer data to a printer. It sends text data to the default or specified printer, or prints the contents of a file to the default or specified printer.

3.4.2.1 Remote Queue Administration

The information from Process Management is not available at this time and will be provided in an update to this manual.

3.4.2.2 Printer Configuration/Administration

The information for Printer Configuration/Administration is not available at this time and will be provided in an update to this manual.

3.4.2.3 Support for Remote LAN Printing

The information for Remote LAN Printing is not available at this time and will be provided in an update to this manual.

3.4.2.4 Print Service Segments

The information for Print Service Segments is not available at this time and will be provided in an update to this manual.

3.4.3 Process Management

The information for Process Management is not available at this time and will be provided in an update to this manual.

3.4.4 Session Management

The information for Session Management is not available at this time and will be provided in an update to this manual.

3.5 Security Administration

Security of the operating system and COE applications including the development environment is necessary to maintain the integrity of applications and data. The DII COE has established an overall scheme for maintaining overall security and the separation of classified and unclassified data. This scheme is outlined in Section 7.1 of the *DII COE I&RTS*. The physical environment in which the system resides is mission-dependent and the responsibility of the individual facility. The following briefly outlines the security measures incorporated into the DII COE and includes on-line system services.

3.5.1 Account Groups

All COE operating environments possess the concept of account groups that are aggregates of users with similar or identical requirements on system resources. Users are normally assigned individual login accounts with configuration files that establish a runtime environment context. These configuration files must be set up and established for each user of the system. An account group segment is a template used within the COE for setting up individual login accounts. Account groups contain template files for defining what COE processes to launch at login time, what functions are to be made available to operators, and preferences such as color selection for window borders. Account groups are described further in Chapters 2 and 5 of the *DII COE I&RTS*.

Account groups can be used to perform a first level division of operators according to how they will use the system. Within an account group, subsets of the available functionality can be created. These subsets are called *roles*. A user may participate in multiple account groups with

multiple roles, and can switch from one role to another without the need to logout and log-in again. Although other groups may be added as needed, five distinct account groups have been identified within the COE:

Privileged User Accounts - COE operating systems provide a 'super user' account to allow knowledgeable System Administrators to perform special functions and to troubleshoot. The COE design philosophy severely restricts the use of these types of accounts. A complete discussion of these accounts in the COE can be found in Section 2.4.1 of the *DII COE I&RTS*.

System Administrator Accounts - The System Administrator Account Group is a specialized collection of functions that allow an operator to perform routine maintenance operations and load segments.

Database Administrator Accounts - The Database Administrator Account Group is used for performing routine database maintenance activities such as backups, restores, archives, and reloads.

Operator Accounts - An Operator Account is designed for application system users performing mission specific tasks such as creating and disseminating Task Orders (TOs), preparing briefing slides, performing *ad hoc* queries of the database, participating in collaborative planning, etc. Limitations on this group are contained in Section 2 of the *DII COE I&RTS*. The precise features available depend upon which mission application segments have been loaded, and the profile assigned to the operator through the Security Manager.

Security Administrator Accounts - Security in the COE is implemented through a Security Server and through a special trusted security client. This client is the Security Administrator application, which is an interface to the Security Server. It allows a Security Administrator to monitor and manage security. Precise functionality of the Security Service is platform-dependent because different approaches have been taken by different vendors to provide security. Available functions include the ability to:

- create individual login accounts
- create defined operator profiles
- customize menus by operator profile
- export accounts and profiles to other LAN workstations
- review and archive the audit log.

The Security Administrator software provides the ability to describe objects (files, data fields, executables, etc.) which are to be protected from general access. This information is used to create profiles that limit an operator's ability to read or modify files. Applications may query the security software to determine the access permissions granted to the current user. The '*Permissions*' file is the mechanism for segments to describe objects and the permissions to grant or deny access to the objects.

3.5.2 Security Services

The Security Services provide the means to set profile configuration, create or edit local and global user profiles, and create or edit local and global user accounts. The security administrator's account group sets the security administrator's environment to execute the profiles and accounts. The Security Manager provides a facility to update internal profile and user account data structures through command line programs. UNIX-based COEs provide additional functionality over the basic operating system as follows:

Role Management - Controls access to applications through the use of roles in the Operator Account Group. It provides functionality to add or delete users to or from a specified role, add or delete applications to or from a specified role, change the name of a role, list all available roles, or list applications assigned to a role.

Account Management - Controls the access of individuals to the system. It provides functionality to add or delete a user to or from the system, assign or remove a user to or from a role, add or delete application data to or from a user, or list the roles assigned to a user.

Session Management - Manages parameters related to an individual user's session. It provides functionality to obtain or set the current user's role, display the textual explanation for an error message, or display the user's UNIX identification number.

3.5.3 Security Software

The COE kernel also provides additional software to supplement the basic security provided with the operating system for a given platform:

Screen Locking - Prohibits input from workstations connected and logged on which are not attended.

Secure Window - Opens a secure, read only console window to display classified output sent to the standard output stream (stdout).

Display Manager - Manages a collection of X-Window displays, prompts for a user ID and password, performs authentication and running of a "session."

Password - Allows users to change their password in the desktop environment.

3.6 Distributed Computing Environment

Distributed Computing Environment (DCE) is an open systems solution that allows users to distribute computing across systems to create a faster, more efficient processing environment.

DCE is based on IEEE POSIX standards (PASC P1003.x). The future of DCE technology is set through open technical forums of the Open Software Foundation (OSF).

The services provided by the COE DCE are shown in Figure 3-1. Core DCE services are provided for all platforms in the DII COE kernel. Extended services vary across platforms and are discussed in the Section 5.0, DII COE Infrastructure Services.

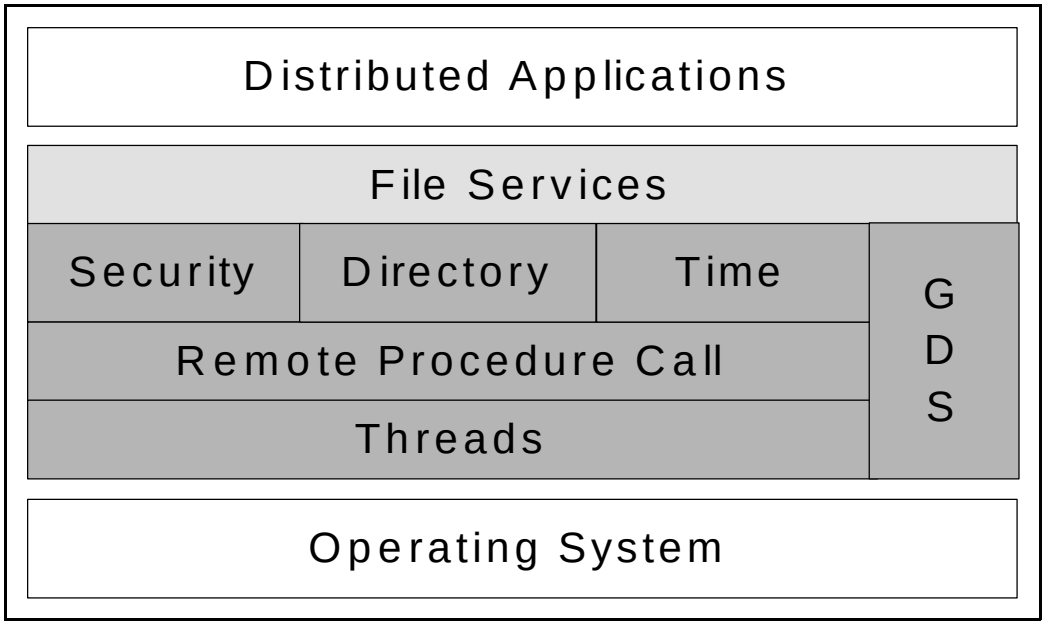

Figure 3-1. Distributed Computing Environment Services

3.6.1 Distributed Computing Environment Cells

A DCE 'cell' is a collection of users, resources and/or services. Typically within a cell, users make use of computing resources. The resources are provided by the servers. One or more servers may implement a Service. A cell may be a collection of machines running DCE services. There may be 3 to 15 Security, Directory, and Time Servers within a cell and approximately 150 clients per server. This is equivalent to having 0 - 25,000 entities per cell. Machines within a cell represent an administrative domain.

3.6.2 Distributed Computing Environment Services

The core DCE provides the following set of services:

Threads - Allows independent execution threads to be created within the same program, allowing for parallel execution of multiple computing tasks with minimal overhead.

Remote Procedure Call - Extends the parallelism provided by threads to allow a procedure to invoke a procedure on another computing resource in a cell. Remote

Procedure Call is a facility for the activation of remote processes in a remote system over communications services. Exception handling is built-in while the complexities of network communications are hidden.

Security Services - Provides for a 'Single,' network-wide registry with the ability to support single sign-on capability for security, database access, and verification of credentials. Service is scalability through the use of replication, client caching, and inter-cell operations. It provides robust services by way of Authentication through Privileged Attribute Certificates and POSIX Access Control Lists. This is done both using intra-cell and inter-cell security mechanisms. The service also provides for message encryption services using Data Integrity and Privacy. It supports powerful delegation support capabilities.

Time Services - Provides a service essential for application synchronization and time value standardization in a controlled transaction environment. It provides scalability through a Hierarchical propagation scheme, a Fault tolerant fall-back configuration, and can Import time from a variety of sources such as the National Time Pendulum (NTP), World Wide Web (WWW), Spectracom, and Global Positioning System (GPS).

Directory Services - Implements a distributed information repository about objects in the computing environment through both the Cell Directory Service and the Global Directory Service. Directory services use both the domain name service (DNS) and X.500 directory services. Attributes are available which describe users, computers, and available distributed services.

3.6.3 Distributed Computing Environment Functions

A 'Client Binding' API is currently available and described in Section 4. It locates a server following DII COE guidelines and provides authenticated access. The guidelines and the DCE functions are described in the *DII COE I&RTS*.

4. DII COE Developer Toolkit and API's

This section gives an overview of the COE Tools and Application Program Interfaces (API). It includes the main functionalities of these tools and their basic uses. More detailed information and descriptions of each Tool and API may be found in the *DII COE Programmer's Reference Manual.*

4.1 COE Tools

One of the strongest features of COE is its wide collection of tools. These tools can be subdivided into two categories; 1) the runtime tools which are delivered in the kernel with the software code, and 2) the development tools which are software tools for developer's to use. The runtime tools are described in Section 3, DII COE Kernel. The developer's toolkit is used to perform the following functions:

- build installation segments,
- test the segments, and
- manage segments.

Appendix B of the *DII COE Programmer's Reference Manual* provides the list of COE Tools and APIs.

4.2 Application Program Interface

APIs provide software applications access to the platform services or to the COE component services. A programmer invokes specific services in their application software by using API calls. Standards-based API calls as specified in the DII COE provide a level of application portability and independence from the underlying platform services.

4.2.1 Concept

A common definition of an API is:

> An API specifies the mapping between program syntax and the features of a specific service, and thereby provides access to that service from applications written in a particular programming language, when the application is bound to the service by the language implementation. (IEEE - PASC POSIX)

An alternate short definition of API is; "an interface is a boundary between two entities for purposes of processing data." For example, an API is the interface between mission application software and the underlying application platform services. The platform is defined as a set of organized resources and services on which the application software will run, and provides specific services required by the application software.

An aspect of the use of APIs is related to the way a service is called. The API call and attributes specified must be bound to a programming language and linked to the program in an executable mode in order to work. A specific language, such as COBOL or C, can have native verbs that call a macro function of the system or network service desired. This is the easiest way to develop programs for the business user. However, a verb or procedure call function from a supplier's product line may lock the program in that environment. Therefore, the API calls (verbs) within programming languages must also be standardized and tested to allow the program to be portable across platforms.

Currently high-level APIs, protocol-specific verbs, or calls for programming languages that will perform the functions across platforms are not available. However, work is ongoing to make the APIs more user-friendly, allowing application programmers to use macro services as utilities in developing new kinds of programs.

It is absolutely vital that developed applications access services only through common COE-approved APIs. The use of private or proprietary (i.e., undocumented or not COE-approved) APIs make component upgrades difficult and cause mission applications, that use those private APIs, to fail or to malfunction.

4.2.2 Role of APIs in Application Development

The access to systems and network services by DII application segments or other mission applications requires the use of program service calls using APIs. In application development, developers use APIs to request the services of the common system components such as directories, file transfers, E-mail, Remote Database Access, etc. The service calls provide access to the specific Application Service Entities (ASE) that are accessible by the appropriate application program linkages to the programming languages' binding services. The APIs are the means of accessing the service elements of all DII network and platform services.

The Developer's Toolkit of the DII COE includes APIs to access services to:

- Help in the development of new applications.
- Help developers provide information to users and solicit input. These are intended to assist when new applications (segments) which have a broader applicability for developers are installed or removed from a system.
- Print data.
- Provide mapping functions.
- Use the Distributed Computing Environment (DCE) for building distributed applications using a client/server model.

The standards-based APIs are documented in the *DII COE Programmer's Reference Manual* and the various DII COE documents that are part of the COE Toolkit.

The APIs for the Common Desktop Environment (CDE), Oracle, and Sybase are not included in the toolkit, but are provided, supported and documented by their respective vendors/developers.

The Developer's Toolkit is provided separately, **not** as a DII COE-compliant segment. Details on how to load the tools onto a DII COE-based system are provided in the toolkit documentation provided in the DII COE documentation, *Installation Instructions for the Developer's Toolkit.*

4.3 Standards

The DoD has adopted existing standards and developed special standards that apply to the COE. These standards and the goal of portability affect the development of applications designed to run in the COE.

The framework for major service areas of the TAFIM applies to the DII COE. Conversely, the DII COE is compliant with the TAFIM's classification of standard services. Refer to Appendix E for a list of TAFIM adopted standards.

The CASE tools and environments specific to COE tools are compliant to the principles of the Portable Common Toolkit Environment (PCTE). PCTE is the ISO/IEC 13719 Standard used with C and Ada Language bindings, commonly referred to as European Computer Manufacturers' Association (ECMA) 149 (PCTE).

Additional standards applicable to this document are related to the software development environment described in the Institute for Electrical & Electronic Engineers (IEEE) 1209 (Evaluation and Selection of CASE Tools), DoD-STD-1467 (Software Support Environment), NIST-ECMA 500-211 (Reference Model for SEE Frameworks), and MIL-HDBK-782 (Software Support Environment Acquisition).

With respect to software life cycle processes, the MIL-STD-498 *Software Development and Documentation*, and the ISO/IEC 12207 *Software Life Cycle Process* standards are applicable. For the GUI behavioral design the MIL-STD-1801 *User Computer Interface* applies.

Standards for APIs are primarily based on Portable Operating Services Interfaces (IEEE POSIX P1003.n series). Additionally API standards created by various industry and consortia are adopted as "Public" APIs. These are X/Open (XPG4 Guide), Open Software Foundation (OSF/DCE), Network Management Forum (NMF) and other expert groups such XAPIA that feed the API standards process into the formal standards developing organizations.

"Look and Feel" standards, contained in the *DII COE Style Guide*, are required for the development of a desktop application. All software is expected to comply with the *DII COE Style Guide* specifications, with deviations occurring only when called for by operational requirements and approved by DISA.

This page intentionally left blank.

5. DII COE Infrastructure Services

DII COE Infrastructure Services are a set of capabilities provided to support mission applications and systems operations on an as-required basis. Infrastructure Services expand and supplement the basic services provided in the Kernel and are installed at the discretion of a site administrator. The *DII COE Programmer's Reference Manual* contains information about each of the applications discussed in this section. Infrastructure Services of both commercial and government products and are usually not available (or tested) on all DII COE platforms, especially Windows@ NT. The *DII COE Programmer's Reference Manual* and applicable *Version Description Document*s should be checked for availability.

5.1 Management Services

DII COE management services extend the basic Kernel capability for system administrators to manage the ongoing operation of their system. These services tend to be commercial packages designed to extend services on a standard UNIX platform. They have been adapted and tested for use within the COE.

5.1.1 System Management Services

System management services provide a comprehensive set of services for administering systems, expanding system administrators' effectiveness by permitting common system administration functions to be performed remotely. It simplifies management of a client/server network of heterogeneous workstations and other desktop systems. The COE provides Tivoli's Management Environment which is compliant with the Object Management Group's Common Object Request Broker Architecture (CORBA). The COE provides Tivoli for both clients and servers that consist of the base Tivoli and extensions.

5.1.1.1 Management Environment

The Tivoli Management Environment is a basic set of system administrator tools.

Administrator management - Provides a mechanism for explicitly identifying system administrators and the system administration resources the administrator is allowed to access, as well as the privileges allowed when accessing those resources. System administrators can perform administration functions without requiring the root password.

Notification groups - Provides a log of administrator activity separated into application-specific notification groups. These notices provide an administration-specific audit trail.

Profiles and profile managers - Manages common system resources (e.g., user accounts, host namespace, etc.) according to a resource template, called a profile. A profile includes the definition of rules that will be enforced for the profile attributes, such as the user password criteria.

Basic managed node - Permits Tivoli to perform remote management actions on client nodes, including querying its physical and logical configuration without having to log into that node.

Policy regions - Establishes default policy, invoked when a new instance of the resource is created (e.g., user). Performs policy validation, used to check that existing resources still comply with the rules.

Interconnection - Interconnects managed resources to provide scalable system administration and to reduce the effects of individual management server failure.

Tasks and jobs - Define administrative procedures for one or more platforms, permitting common system administration tasks, such as removing core files, pruning out unused user accounts, or checking for invalid user passwords, and perform security audits.

Scheduled actions - Perform certain actions at regular intervals or at off-hours so that they do not interfere with mission-critical operations.

5.1.1.2 Administration

Extends the basic Tivoli profile capability to provide user, group, and host namespace management. Tivoli Admin also provides basic and extended Network Information Service (NIS) domain management. Tivoli Admin's main functions are as follows:

User account management - Supports adding, editing, and deleting user accounts, as well as distributing user account information to other profile managers, to managed nodes, and to NIS and Extended NIS (NIS+) domains.

Group management - Supports adding, editing, and deleting UNIX groups, as well as distributing group information to other profile managers, to managed nodes, and to NIS and NIS+ domains.

Host namespace - Supports adding, editing, and deleting host name - Internet Protocol (IP) assignments, as well as distributing host namespace information to other profile managers, to managed nodes, and to NIS domains.

NIS - supports adding, editing, and deleting NIS maps and map data, as well as making and pushing NIS maps.

NIS+ - supports the distribution of user and group profiles in a NIS+ environment.

5.1.1.3 Software Distribution

Tivoli Courier provides software distribution (and removal) across heterogeneous platforms with a wide number of options, including repeater capability to reduce traffic over wide-area networks (local fan-out), single source to many destinations in a single action and pre or post processing.

5.1.1.4 Monitoring

Tivoli Sentry provides a secure, distributed, general-purpose system monitoring capability. It does not require polling to detect changing conditions on clients, as Tivoli Sentry monitors run locally, that is, on the managed node being monitored. Sentry also implements role and privilege definitions to control which administrators are permitted to define or modify Sentry monitors. When specified conditions are detected, Sentry can be configured to take corrective action automatically, send e-mail, notify specific administrators, or run an executable.

5.1.2 Network Management Services

Network management services provide capabilities for an authorized user to manage server and network activity from a client computer.

5.1.2.1 Management Information Base Functionality

The Internet community has defined sets of standards for networking which compose various Management Information Bases (MIBs). These standards start as Requests for Comments (RFCs) and consist of definitions of measurable quantities or items of information. These quantities are broken down into groups. Some of these standards concern quantities useful in a Network Management function. Network Manager Systems implement these MIBs by collecting and monitoring these information sets in addition to setting specifically identified elements.

5.1.2.1.1 Host Resources MIB

The host resources MIB consists of 6 groups of information designed for managing host systems on a network:

System Group - Information relating to system startup, system time and running time, and numbers of users and processes.

Storage Group - Parameters relating to local storage areas (e.g., disk partitions and file systems) and current usage information.

Device Group - Identifies and provides status on all installed devices, processors, ports, disk drives, network interfaces, etc.

Running Software - Identifies applications currently running on the system and provides details including type of process, calling sequence and parameters, path, and status.

Running Software Performance - Provides processor time and memory being used to run the application.

Installed Software - Identifies (name, description, version, type, and date installed) software packages installed by the COE segment installer. Also, identifies the current version of the operating system and installed patches.

5.1.2.1.2 Network Management MIB

The Network Management resources MIB consists of 10 groups of information that constitute a simple architecture and system for monitoring Transmission Control Protocol/Internet Protocol (TCP/IP) based networks:

System Group - Information on network entities including an identification of the network management subsystem, a description, network name, location, contact, and provided services. The time since the last restart is also provided.

Interfaces Group - Detailed information about each network interface in a managed network entity including a description, status, performance, current throughput, and error rate.

Address Translation Group - Functionality transferred to other groups. Retained for compatibility with earlier standards.

IP Group - Detailed information regarding entity processing of IP datagrams, a table containing the entities' IP addressing information, and a table containing information for each network route known to the entity, and a table to map IP addresses to physical addresses.

ICMP Group - Detailed information regarding entity processing of Internet Control Message Protocol (ICMP) messages.

TCP Group - Operating parameters of the entity and dynamic information regarding any open TCP connections.

UDP Group - Detailed information regarding User Datagram Protocol (UDP) datagrams and a table of UDP Listeners (processing application) accepting UDP datagrams.

EGP Group - Information regarding Exterior Gateway Protocol (EGP) messages and a table of the entities EGP neighbors.

Transmission Group - Detailed statistics dependent on the physical method used to transmit information.

SNMP Group - Detailed information on the processing done by a Simple Network Management Protocol (SNMP) entity.

5.1.2.1.3 Remote Network Management MIB

The Remote Network Management MIB extends the Network Management MIB to allow for the remote management of a network. The MIB allows a probe to collect statistics continuously even when communication with the remote manager is not possible or efficient. The probe will attempt to notify the manager when exceptional conditions exist. The MIB consists of 9 groups of information that define objects for managing remote network monitoring devices:

Statistics Group - Statistics measured by probes for each monitored interface.

History Group - Periodic statistical samples from the network stored for later use and to provide an operational baseline.

Alarm Group - Statistical samples of probe variables that are compared to thresholds to signal significant deviations from normal operations.

Host Group - Statistics associated with host systems identified on the network.

HostTopN Group - Prepares reports describing hosts that top a list ordered by one of the available host parameters.

Matrix Group - Statistics for conversions between sets of two addresses.

Filter Group - Processes captured packets that match a filter equation. Captured packets can generate events.

Packet Capture Group - Permits the capture of packets after they flow through a channel.

Event Group - Generates events and provides notification of events from the monitoring device.

5.1.2.2 COE Functionality

The COE currently provides 3 commercial packages for network monitoring which supplement the basic functions provided by the Kernel:

Empire System Manager Agent - Implements the Network Management and the Host Management MIBs. Additionally a UNIX-specific extension to the Host Management MIB is implemented.

NetMetrix Remote Monitoring Agent - Implements the Remote Network Management MIB and a toolbox containing a graphic toolset.

Streams - Provides a general facility and a set of tools for the development of UNIX communications services.

5.1.3 Print Management Services

Print management services provide enhanced functionality for network printers. The COE includes a SUN product, Newsprint, which provides network printing support and utilities and print file filters and fonts.

5.2 Communications Services

COE communications infrastructure services provide the capability to interact with other DoD message and data systems.

5.2.1 COE Communications Software / USA Comm Server

The COE Communications Software (CS) provides reliable message delivery through Tactical Communications Interface Modules (TCIMs), Tactical Radio, LANs and WANs. The Communications Server supports both DCE and non-DCE network environments. The CS enables clients without TCIMs to send/receive messages through TCIMs attached to other servers. The COE CS provides the common communications infrastructure for the Army Common Hardware and Software program.

5.2.2 Unified Build (UB) Comms Service

The Comms service takes coded input from various tactical and message traffic inputs and decodes the input for use by other UB services. Input may be from Link 11, Link 14, navigational (GPS type) interfaces, and other special interfaces. These services are discussed in Section 6.4, DII COE Support Applications, Correlation.

5.3 Data Management Services

DII COE data management services provide an application-independent capability to maintain and administer data. In the DII COE, these services are provided by a COTS database management system (DBMS). The *DII COE I&RTS* and the *DII COE I&RTS Database Development and Integration Standards* documents discuss database concepts and the DBMS design environment for the DII COE client-server architecture.

5.3.1 COE Functionality

COE DBMS products support ANSI standards for Structured Query Language (SQL). Additionally, COE DBMS's must provide additional functionality for inclusion in the COE.

5.3.1.1 Distributed Databases

Due to the nature of many military systems operating on the COE, data may be spread across multiple sites. Therefore, COE DBMS products must support distributed databases. Data is replicated in a distributed database when copies of particular objects or records exist in more than one of those sites. Data is fragmented when they are split among sites. Databases are distributed (fragmented or not) to improve responsiveness and increase availability in systems that serve geographically-dispersed communities. In some circumstances, databases are replicated to enhance their survivability. The implementation objective of any distributed database is to provide location transparency. This means that the user need not know where data reside to enable access to data or to process the data.

5.3.1.2 Data Integrity

A COE DBMS must provide features that support data integrity to protect the information stored within a DBMS. To preserve database integrity, the database server must:

- Prevent connections between an application and data that belongs to any other application. COE-based systems are typically secure systems that contain and process classified data. The database management component must conform to the security policies and practices of the overall program. Otherwise, the database server supports the data access restrictions and integrity assumptions incorporated in each database.

- Ensure the recoverability of failed transactions or of a crashed system. The "ACID" properties, for atomicity, concurrence, isolation, and durability, of database transactions are the responsibility of the applications accessing the server. However, supporting these transaction properties is the responsibility of the server. Developers must pay special attention to transaction isolation due to the multiple database configuration of most COE-based systems.

- Enforce the developers integrity constraints when they are defined within the database. Database developers are responsible for defining and implementing the integrity constraints of their databases. Application developers must ensure that their applications connect properly to their databases and do not connect improperly to anyone else's databases. Adoption of these practices protects all applications' data and allows the database server to maintain all databases reliably.

Within a component database implementation of data integrity takes the form of constraints and business rules. In the current context, constraints are defined as the rules within the database that govern what values may exist in an object. Business rules are those rules within the database that govern how data is updated and what actions are permitted to users.

Commercial database management systems were traditionally limited in their ability to support the variety of constraints and business rules needed in a database. This resulted in coding the required

constraints and business rules into the applications, rather than in the database where they should be. The federated database architecture and the applications that maintain those databases are developed independently. This causes difficulty in ensuring uniform and consistent enforcement of those rules and constraints.

Henceforth, developers shall place their business rules and constraints in their databases rather than the applications. This will retain control of data maintenance access with the developers where it belongs and ensure that constraints cannot be bypassed. Developers have the knowledge of their constraints and business rules; DBAs and users do not.

Figure 5-1 shows an example of applications and database independence by placing constraints into the data access rather than in the applications. The placement of business rules and constraints in the database promotes client/server independence. The efficient implementation of constraints and business rules makes maximum use of current DBMS capabilities. When the rules are in the component database, the application is less dependent on the COTS product. This approach reduces network communications loading by allowing the DBMS, rather than the application, to enforce the rules. Checking rules within the database avoids passing multiple queries and their results over the network between the DBMS and the application.

5.3.2 Database Management Systems

The COE currently supports 3 DBMS's. Although each commercial product has some unique features, use of these features is discouraged since it does not support application portability. The COE is working toward a shared data environment. Currently the COE includes the following commercial products (check for availability on specific platforms):

- Oracle
- Sybase
- Informix

5.4 Network Services

The COE supplements the Kernel network services available to the user. Currently the COE implements DCE extended services that build on the DCE core provided in the Kernel and a suite of Internet/World Wide Web applications.

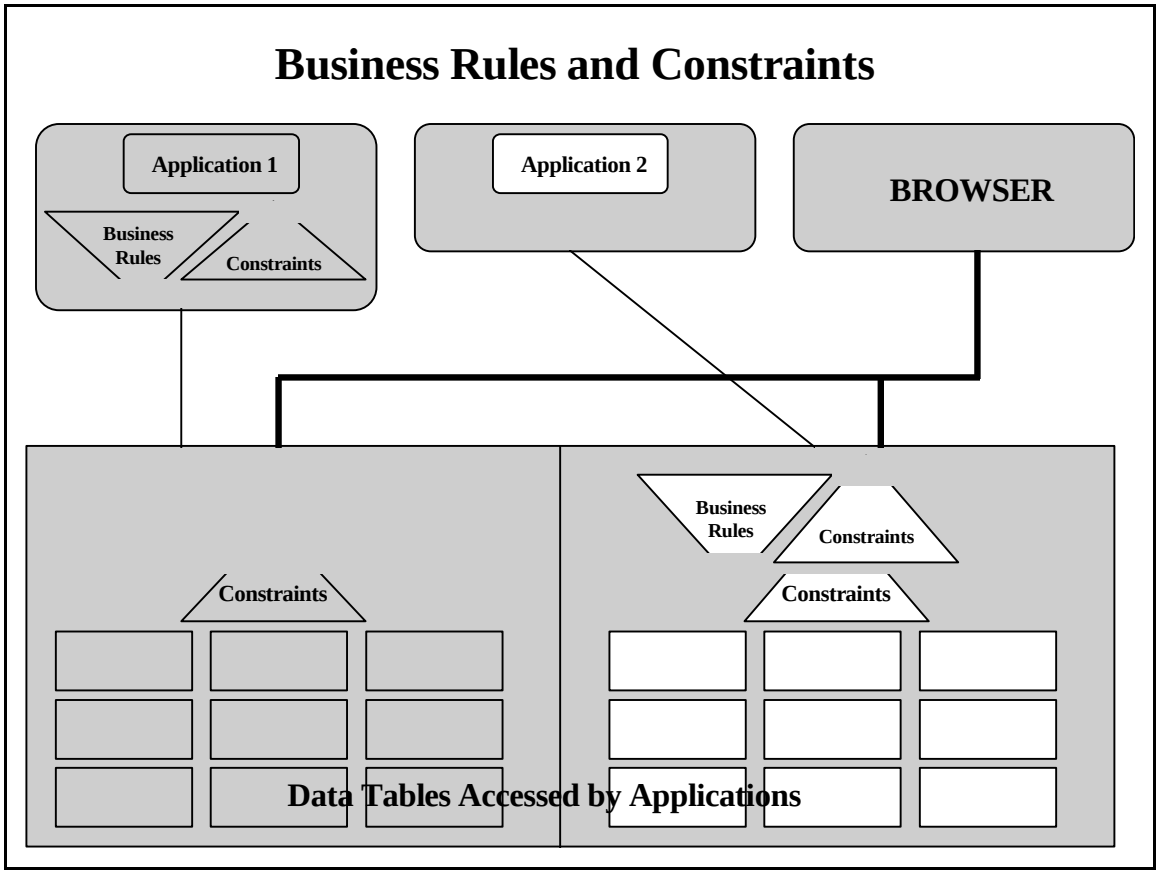

Figure 5-1. Business Rules and Constraints

5.4.1 Distributed Computing Environment

The concept of the Distributed Computing Environment was introduced in Section 3 since the client portion of DCE resides on every workstation. However, to configure a DCE cell and activate a distributed environment, one or more servers must be configured. This application is contained in the infrastructure utilities. At least one server is required to support the DCE. The server directly supports the Cell Directory Service (CDS), the Security Service, and the Distributed Time Service (DTS).

5.4.1.1 Cell Management

The COE also provides a utility which enables graphical management of a DCE cell. The DCE Cell manager by Chisholm Technologies provides the following set of tools:

Namespace Manager - Views, searches, and edits the CDS.
Security Manager - Views, searches, and maintains the security registry.
Configuration Manager - Monitors and manages DTS and host status.

5.4.1.2 Distributed File Service

The Distributed File Service (DFS) is an enhancement to the DCE built on top of the core DCE services provided in the Kernel. DFS extends the core to support file sharing and data management activities. DFS enables collections of workstations to share files as a single unit.

5.4.2 Internet Services

Access to the Internet has become an essential mission requirement for many organizations. The COE provides a set of utilities to launch the user into the Internet and the World Wide Web:

Web Browser - HTML-capable World Wide Web browser from Netsacpe.

Web Server - Utilities for establishing and maintaining home pages and a World Wide Web server from Netscape.

News Server - Utilities for establishing and maintaining discussion groups on a server from Netscape. Discussion groups can be private (designated members) or public.

News Make Group - Information is not available at this time but will be provided in a later release of this manual.

5.5 Utilities

Utilities provide an additional set of application-independent services in support of DII COE mission applications, infrastructure services, and authorized users.

5.5.1 General Utilities

General utilities provide generic capabilities to COE platforms.

5.5.1.1 File Related Utilities

The COE provides basic utilities to provide enhanced functionality for users in obtaining and storing data.

File Transfer Protocol (FTP) - General purpose utility that allows transfer of files to or from remote computers. FTP can be set up as anonymous or require a specific user logon sequence.

File Compression - General purpose file compression scheme (GZIP) which reduces the space used by a data file without losing any information contained in the file. Similar to the ZIP function used on PCs.

5.5.1.2 UNIX Script Utilities

UNIX script utilities are interpretive languages that provide additional functionality over standard UNIX command line utilities

Practical Extraction and Report Language (Perl) - Perl is optimized for scanning text files, extracting information, and generating reports. Primarily used for systems management by systems administrators.

Tool Command Language (TCL) - TCL is a script language with aspects of procedural languages with access to an associated graphical tool kit. Used for simple programs which require a graphical user interface.

5.5.1.3 Windows Emulation

Windows Applications Binary Interface (WABI) is a Sun solution to integrating MS Windows@ applications into the Solaris environment. It requires the purchase of MS Windows@ 3.11 and the actual application. WABI does not guarantee that all applications will operate correctly. Information about how to check WABI/application compatibility is in Section 6 of the *DII COE Programmers Reference Manual*.

5.5.2 Security Utilities

COE security utilities fall into two broad classes, probes that assist the security administrators in preventing unauthorized access and sentries that attempt to prevent or detect unauthorized access.

5.5.2.1 Probes

Probes are utilities used to detect security weakness and vulnerabilities that can be used to penetrate a system. COE provides the following utilities:

Crack - A compute intensive, configurable utility that attempts to guess passwords. Guessable passwords are exploitable holes in system security. Freeware versions of Crack are available.

Security Administrator Tool for Analyzing Networks (SATAN) - Systematically probes an Internet host for known security weaknesses. SATAN is distributed as Freeware.

Security Profile Inspector (SPI) - A collection of security tools and utilities. Contains tools to attempt to expose weak file or group structures, test operating system vulnerabilities, or crack passwords. Detection tools use check sums to detect system authenticity and patch currency, or changes in files, users and groups. A utility is provided which allows inquiries into security objects. SPI is available only to the government.

5.5.2.2 Sentries

Sentries are usually passive routines that detect the actions of a probe or a system penetration. The COE provides the following utilities:

Courtney - Released by the government coincident with the release of SATAN to detect possible SATAN probes on a host. Courtney is available in the public domain.

TCP Wrappers - Monitors external requests for common network services such as FTP or TELNET. Can provide additional security by limiting these services to specific workstations. Shareware versions of TCP wrappers are available.

Tripwire - Detects possible intrusions and assesses damage by comparing electronic file signatures. Tripwire is also public domain.

6. DII COE Support Applications

DII COE support applications are an upper level layer of applications software and toolkits designed to provide specific functionality to an end user or to support the development of an application or process. Support applications are tailored to user functions which will be performed at a workstation. Support Applications consist of commercial and Department of Defense (DoD) products. Availability of applications may vary according to hardware platform. Additional details on availability and functionality are contained in the *DII COE Programmer's Reference Manual*.

6.1 Office Automation

Office automation is a collection of general purpose desktop capabilities that enhance end user productivity. Common office automation applications include:

Word Processor - Provides document preparation and formatting capabilities that exceed those found in the desktop editor. The service allows revising, advanced formatting, and printing of documents, papers, and reports. It also interfaces and incorporates the products of other office automation components.

Spreadsheet - Processes numerical and textual data that can be arranged in sets of rectangular arrays. The service includes formulae which may reference other cells in the array, provides statistical functions, and presents information in the array in a graphical display.

Presentation Graphics - Generates the electronic "slides" necessary to present briefings. Presentation graphics usually includes a facility for drawing and illustration. The service facilitates planning, writing, revising, manipulating, formatting and standardizing, and printing of a series of textual and graphical images.

Mail Services - Provides an advanced electronic messaging capability to the desktop. The service enables the creation, sending, receiving, viewing, storing, and forwarding of electronic information. Information may be text, audio, video, imagery, graphics, animation, multimedia, or hypermedia and may be annotated in the message or attached. Electronic signature authentication is also provided.

Internet Relay Chat - Provides users the ability to electronically interact with other users in realtime. The service allows sharing of messages between group members as they are typed.

6.2 Mapping, Charting, Geodesy, & Imagery

The Mapping, Charting, Geodesy, & Imagery (MCG&I) service is provided in the Joint Mapping Toolkit (JMTK) software package. The service provides the ability to graphically present information to the operator. The JMTK provides the following functionality:

Manipulate Symbols - Provides the ability to alter the characteristics of an object or a family of objects. Function calls include the ability to:

- Modify an object family's display attributes, data field, fill characteristics, or visibility
- Copy display attributes into a template
- Animate an object
- Retrieve attribute or coverage information about an object
- Set a parent object's display attributes, data field, fill characteristics, or visibility
- Change selected physical characteristics of an object.

Draw Objects - Provides the ability to create and alter graphical objects. Function calls include the ability to:

- Create text, polygon or polyline objects through animation
- Draw an arc, box, character (including 16 bit symbols), circle, ellipse, line, polygon, polyline, rectangle, sector, line segment, slash, symbol, or text string
- Include a bitmap in a drawing
- Alter the orientation of a line segment
- Change the text in textual objects.

Control Display Settings - Provides the capability to alter attributes specific to text objects. Function calls include the ability to:

- Modify an object family's font, line style, line type, or line width
- Apply a template to an object or object family
- Set an object's font, line style, line type, or line width
- Set the hilight color
- Change the color of an object or object family to the hilight color.

Control Display Features - Provides the capability to alter features and symbols on a map. Function calls include the ability to:

- Add a feature or list of features to a map
- Add a point to a polygon or polyline
- Change the color of an object or object family
- Sets whether or not an object or object family is selectable
- Create an object through animation
- Create an object attributes template
- Draw a weather line segment (front)
- Set the overall intensity, foreground color, and background color of a geographic display.

Edit Features - Provides the capability to move an object from a window, remove an object, or modify features on a map. Function calls include the ability to:

- Remove classes of features or products from the library
- Modify attributes of a feature or list of features on a map
- Move an object on a map
- Remove a feature from a map
- Remove a list of map products from a display
- Remove an object from an object list.

Transform Data - Provides the capability to perform conversion to standard units. Function calls include the ability to:

- Extract projection data from a given window
- Determine the latitude and longitude of a point on a geographic display
- Determine the location on a display given latitude and longitude.

Conversion is provided between radians, degrees, nautical miles, miles, kilometers, and feet.

Manage Windows - Provides a consistent Windows interface to the Chart Manager to windows from client applications. Function calls include the ability to:

- Initialize and terminate the interface
- Request ownership of a feature product
- Request rendering responsibility for a map product
- Abort a draw command
- Add an object to a list or class
- Add a product or list of products to a map
- Change displayed maps and features
- Create an object class or list
- Create or terminate a map window
- Erase an object family or list of objects
- Draw a world view map in a window
- Transfer an object to another class or list
- Obtain the X Window ID of a geographic display
- Map a window to the screen to allow display
- Terminate JMTK operations on a window
- Reorder map products in a display
- Hide a window
- Get and register a channel to a window.

Adjust Display View - Provides capabilities to alter the view of an existing map. Function calls include the ability to:

- Magnify the current view (zoom in)

- Center the display on a specified map point
- Change the scale of a map
- Change the mode of the cursor and the left mouse button
- Set the boundary attributes or width of a geographic display.

Query Spatial Database - Provides functionality to use the spatial database (SDB). Function calls include the ability to:

- List available SDBs
- Connect to a SDB
- List available maps
- Define or retrieve the area of interest (AOI)
- Locate search path information related to a specific data type within an AOI
- Manage search path information
- Process SDB matrix data
- Return error messages associated with an error code
- End the connection to the SDB.

Manage Display - Provides general display management services. Function calls include the ability to:

- Abort object animation
- Manage input data sources
- Manage interval timer services
- Transfer buffered requests to the chart manager
- Flush event queues or extract an event from a queue
- Count pending map events
- Change the order within a queue
- Manage point select focus
- Send events to other processes using a window
- Set control keys for animation.

Transform Coordinates - Provides conversion between geodetic, universal transverse Mercator, and military grid reference map coordinate systems. Function calls include the ability to:

- Translate coordinates between the 3 systems
- Translate coordinates to and from text.

Manage Symbol Library - Provides the capability to use reference map features. Function calls include the ability to:

- Operate map reference lists
- Describe the list of features drawn into a map window.

Terrain Analysis - Provides capabilities to make tactical decisions based on the geographic display. Function calls include the ability to:

- Determine area of coverage for weapons fanning
- Determine line of site characteristics.

Process Errors - Provides error processing capabilities for the Chart Manager and draw module components of the JMTK. Function calls include the ability to:

- Process chart manager errors for clients
- Handle errors that occur in the draw module
- Indicate to the chart manager that an error has occurred in a specific window.

Provide Information - Allows end users to extract information from a geographic display. Function calls include the ability to:

- Provide the distance and bearing between 2 points
- List available chart manager features
- List display features or attributes for a map window
- Extract information about a map window.

Manipulate Symbols - Provides Chart Manager parsing routines. A function call returns the next matching option on a command line.

Communicate with Windows - Manages the communications channel to the Chart Manager. Function calls include the ability to:

- Identify features or products to the draw module
- Attach a draw module to a chart manager
- Initialize a draw module, communicate between the chart manager and a draw module
- Manage a communications channel
- Interface with Xwindow
- Provide socket information about a communications channel
- Manage events
- Terminate chart manager execution.

Manage Memory - Provides capabilities to manage memory and search functions. Function calls include the ability to:

- Obtain and release memory
- Provide the current map search path.

6.3 Messaging

Message Handling capabilities in the COE are provided by the Common Message Processor (CMP). It provides the functionality required to prepare, receive, analyze, and validate United States Message Text Format (USMTF) and Army Variable Message Format (VMF) messages. The CMP is also able to normalize message data into formats usable by other applications.

6.3.1 User Services

The CMP provides a user interface to access message handling functions and built-in journaling functions. Messages can be interactively created and edited in a distributed environment. The CMP's analysis capabilities support the filtering of message traffic and assigning them to separate windows based on parameters extracted from the message such as recipient, time stamp (date-time group), precedence, and classification. Messages may be generated automatically from operational databases.

6.3.2 Application Support

The core of the CMP is in two standalone software systems which handle the two basic aspects of message handling. Additional modules perform support functions. Rules for message processing are contained in a set of tables that must be configured. CMP provides the following functionality:

Message Processing - Analyzes and controls the processing of inbound message traffic received from Communications Services (see Section 5.3). Function calls are available to obtain an inbound message, determine type of processing required, journal a message, profile a message to determine interest level, validate a message to ensure sets and fields comply with rules established for subsequent processing, parse a message to extract information, and pass extracted information to another location or process (i.e., additional processing, database posting, viewing, etc.)

Message Generation - Creates a user or computer-generated message and sends it to the communications services area for transmission. Function calls are available to manually generate a message according to a template, automatically generate a message according to a template using information passed from an automated process, determine detailed addressing data for the message header, provide for electronic coordination prior to message release, and route the completed message to a user with message release authority for dissemination.

Message Handling Support - Performs general support and common services for message processing. Function calls are available to configure CMP components, process errors external to the CMP which impact message handling, audit classification aspects of message processing, normalize specified formats to USMTF, translate to and from VMF, combine multi-section input messages and section output messages, and annotate messages being coordinated.

Message Journaling - Provides an interface to the systems journal of all incoming and outgoing messages. Function calls are available to modify a journaled message and retransmit, search all received messages, and list or display all or selected released messages.

6.4 Alerts

The alerts service builds on the signal processing functions in the Kernel (see Section 3.1) to manage tactical processing on a workstation or, with an alerts server, on a family of workstations. The alerts service defines alert and event message formats and controls the posting, reviewing, queuing, and dequeuing of alerts and events. Events can be generated by a process or generated by a system event. The alerts service provides:

Alert Generation - Provides a function call that instructs the alerts server to generate a specific alert.

Alert Definition - Provides an interactive interface to create and name routing definitions that include addressing by role, duty group, or user ID.

6.5 Correlation

The Correlation service is part of the Unified Build (UB) software package. The service maintains a consistent tactical picture across a theater of operations based on inputs provided by service components. The service associates high-level tactical objects (ships, submarines, and aircraft), fixed and mobile installations, and land force units with tracking data. Tracking data can be obtained from Electronic Intelligence (ELINT) sources, Tactical Digital Information Links (TADIL), and from Joint Surveillance Target Attack Radar System (JSTARS) type Ground Moving Target Indicator (GMTI) radar tracks.

6.6 Situation Display

The Situation Display service is part of the UB software package. The service displays the current tactical picture across a theater of operations based on the information stored in the Track Database. The track data, representing correlated and uncorrelated tactical elements, is presented on graphic situation displays and made available for tactical decision aids that support decision making.

This page intentionally left blank.

7. DII COE User Information

This section provides basic user information and covers the DII COE standards in terms of applicable reference documents available to the COE users. A complete bibliography of DII COE documentation is available in Appendix C.

The DII COE reflects the principles, architecture and standards adopted by TAFIM which includes the Portable Common Toolkit Environment (PCTE) and the ongoing work in the Integrated Software Engineering Environment (ISEE) that are also related to the Portable Operating Systems Interface (POSIX). Appendix E of this manual covers the list of Adopted Information Technology Standards of the TAFIM.

7.1 POSIX Calls

POSIX originated in the Institute of Electrical and Electronics Engineering (IEEE) standard committee for the Portable Operating Standards Committee (PASC) and refers to a Portable System Interface for compliant computer environments.

There are three parts to the implementation of the standard:

- Definition of terms, global names and concepts
- Interface facilities
- Data interchange format.

A list of POSIX function calls (APIs) appears in Appendix D of this reference manual. They are part of the interface and are required for use in the programming environment to effect portability to assure that programs can interface reliably with any POSIX-compliant operating system using the calls specifications.

7.2 Reference Documentation

Refer to Appendix C for a complete list of reference documents. The following standards are applicable directly to the DII COE.

- POSIX.1 is the Portable Operating System Interface that is currently a Federal Information Processing Standard (FIPS) 151-2 based on ISO/IEC 9945-1 international standards and born from IEEE POSC P1003.1 (ANSI) Standard.

- POSIX.2 is part 2 of the Shell and Utilities is currently the FIPS 189 based on ISO/IEC 9945-2 international standard and born from IEEE POSC P1003.2 (ANSI) Standard.

7.3 Extending the COE

Some segments may require extensions to the COE or extensions may be desired to obtain additional functionality. In this case, refer to Section 5.7 of the *DII COE I&RTS*. It describes techniques for making desired common extensions to the COE, such as adding menu items or icons to the desktop. It also describes techniques for making common types of extensions and techniques for addressing less frequently encountered extensions that may be necessary in the user's environment.

7.4 Problem Reporting

DISA has established a procedure to report any problems detected in the DII COE and has established an Incident Control Center to process incoming problems. Problems are either recorded on a Global System Problem Report (GSPR, DISA form 291) or transmitted via electronic mail.

Once received, the DISA Configuration Management (CM) department assigns a control number to the problem and provides the number to the originator prior to document review. The problem is then sent to a system engineer for evaluation. If the problem has been fixed, CM is notified and the update is distributed. Otherwise, the problem is assigned for action. When completed, CM is notified and the updates propagated as before.

The procedure for end-user notification of disposition of the problem reported is under consideration and will be published when established and approved for dissemination.

7.5 COE On-line Services

The DII COE also features a look ahead at the future of the technology including new development tools by providing a set of on-line services. These Internet- and intranet-based services allow rapid communication and development of strategic plans, requirements analysis, and documentation. They also help in the timely development of, dissemination of, and enterprise-wide access to information and segments. These services offer a wide variety of information-representation, viewing, and processing functions across the major corporations, governmental agencies, and global enterprises making up the COE community. For more details on the COE on-line services, see Section 7.0 of the *DII COE I&RTS*. To obtain information on-line consult the Internet Website addresses (Universal Resource Locator (URL)) in Appendix D.

7.5.1 Security Features

The COE on-line services can be divided into classified and unclassified systems. The classified system is accessible only via the Secret Internet Protocol Router Network (SIPRNet), while the unclassified system is openly accessible on the Internet. Both types of systems use secure operating systems, databases, and network software and maintain auditing of all system access and of all sensitive operations. Classified and unclassified components reside on physically distinct computer systems. The classified system is further protected by three layers of security

consisting of firewalls, user authentication, and encryption. For more details on security features, see Section 7.1 of the *DII COE I&RTS*.

7.5.2 COE Information Server (CINFO)

The COE information server (CINFO) provides information to the COE community via the World Wide Web (WWW) and SIPRNet. Non-sensitive general information is provided by the unclassified WWW site. The classified site on SIPRNet contains more detailed information, including a list of all available segments, the segment versions and all segment patch information, including the information on upcoming system changes, and special installation instructions. For more details on CINFO, see Section 7.3 of the *DII COE I&RTS*.

7.5.3 Mirror Sites

Mirror sites are Software Support Activity (SSA) sites maintained by services or agencies containing local copies of COE on-line services. Mirror sites contribute to the efficiency of configuration management, development, and project communications. Local copies of the on-line services residing on each mirror site are updated periodically. For more details on mirror sites, see Section 7.4 of the *DII COE I&RTS*.

This page intentionally left blank.

Appendix A: Acronym List

AOI	Area of Interest
API	Application Program Interface
APP	Application Portability Profile
ASE	Application Service Entities
C4I	Command, Control, Communications, Computers, and Intelligence
CDE	Common Desktop Environment
CDS	Cell Directory Service
CINC	Commanders in Chief
CINFO	COE Information Server
CMM	Capability Maturity Model
CMP	Common Message Processor
COE	Common Operating Environment
CM	Configuration Management
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off-the-Shelf
CS	COE Communications Software
CSCI	Computer Software Configuration Items
DBA	Database Administrator
DBMS	Database Management System
DCE	Distributed Computing Environment
DFS	Distributed File Services
DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DNS	Domain Name Service
DoD	Department of Defense
DTS	Distributed Time Service
ECMA	European Computer Manufacturers' Association
EGP	Exterior Gateway Protocol
ELINT	Electronic Intelligence
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol
GMTI	Ground Moving Target Indicator
GOTS	Government Off-the-Shelf
GPS	Global Positioning System
GSPR	Global System Problem Report
GUI	Graphical User Interface
HTML	Hypertext Markup Language

ICMP	Internet Control Message Protocol
IDB	Integrated Database
IEC	International Electrotechnical Commission
IEEE	Institute for Electrical & Electronic Engineers
IP	Internet Protocol
IRM	Information Resource Managers
ISEE	Integrated Software Engineering Environment
ISO	International Standards Organization
JIEO	Joint Interoperability and Engineering Organization
JMTK	Joint Mapping Tool Kit
JOPES	Joint Operations Planning and Execution System
JSTARS	Joint Surveillance Target Attack Radar System
JTA	Joint Technical Architecture
LAN	Local Area Network
MIB	Management Information Bases
MCG&I	Mapping, Charting, Geodesy, and Imagery
NIS	Network Information Service
NIST	National Institute of Standards and Technology
NMF	Network Management Forum
NTP	National Time Pendulum
OSD	Office of the Secretary of Defense
OSE	Open System Environment
OSF	Open Software Foundation
PASC	Portable Application Standards Committee
PCTE	Portable Common Tool Environment (ECMA)
PERL	Practical Extraction and Report Language
POC	Point of Contact
POSIX	Portable Operating System Interface for UNIX
PSA	Principal Staff Assistant
RDBMS	Relational Database Management System
RFC	Requests for Comments
RPC	Remote Procedure Calls
PSA	Principal Staff Assistant
SATAN	Security Administrator Tool for Analyzing Networks
SDB	Spatial Database
SEI	Software Engineering Institute
SIPRNet	Secure Internet Protocol Router Network

SNMP	Simple Network Management Protocol
SPI	Security Profile Inspector
SSA	Software Support Activity
TADIL	Tactical Digital Information Links
TAFIM	Department of Defense Technical Architecture Framework for Information Management
TCIM	Tactical Communications Interface Modules
TCL	Tool Command Language
TCP/IP	Transmission Control Protocol/Internet Protocol
TO	Tasking Order
UB	Unified Build
UDP	User Datagram Prococol
URL	Universal Resource Locator
USMTF	United States Message Text Format
VMF	Army Variable Message Format
WABI	Windows Applications Binary Interface
WAN	Wide Area Network
WWW	World Wide Web
XAPIA	X.400 API Association

This page intentionally left blank.

Appendix B: Glossary

Account Group: A template for establishing a runtime environment context for individual operators. Account groups are typically used to do a high-level segregation of operators into system administrators, security administrators, database administrators, or mission-specific operators.

Affected Account Group(s): The account group(s) to which a segment applies. Functionality provided by the installed segment will normally appear to the operator as new menu items or icons in the affected account group(s).

Aggregate Segment: A collection of segments grouped together, installed, deleted, and managed as a single unit.

Application Program Interface (API): The API is the interface, or set of functions between the application software and the application platform. An API is categorized according to the types of service accessible via that API. There are four types of API services: 1) User Interface Services, 2) Information Interchange Services, 3) Communication Services, and 4) Internal System.

Approved Software: Commercial software products that have been tested as compatible with the COE. In this context, approved software implies only that the software has been tested and confirmed to work within the environment. It does not imply that the software has been approved or authorized by any government agency for any specific system.

Bootstrap COE: That subset of the COE that is loaded in order to have enough of an operational environment that segments can be loaded. The bootstrap COE is typically loaded along with the operating system through vendor-supplied instructions or UNIX commands such as *tar* and *cpio*.

Client: A computer program, such as a mission application, that requires a service. Clients are consumers of data while servers are producers of data.

Commercial Off-The-Shelf Software (COTS): Software that is available commercially. Examples include versions of UNIX, X Windows, or Motif, as well as approved software such as Oracle, Sybase, and Informix.

Common Operating Environment (COE): The architecture, software infrastructure, core software, APIs, runtime environment definition, standards and guidelines, and methodology required to build a mission application. The COE allows segments created by separate developers to function together as an integrated system.

Community Files: Files that reside outside a segment's assigned directory. To prevent conflict among segments, community files may be modified only through the COE installation tools. Examples of community files include: */etc/passwd, /etc/hosts, /etc/services*.

Compliance: A numeric value, called the compliance level, which measures the degree to which a segment conforms to the principles and requirements defined by COE standards, and the degree to which the segment makes use of COE services. Compliance is measured in four areas, called compliance categories. The four categories are Runtime Environment, Architectural Comparability, Style Guide, and Software Quality,

Component Database: Individual databases within a multi-database design.

Configuration Control Board (CCB): The organization responsible for authorizing enhancements, corrections, and revisions to the COE, or to a COE based system.

Database: A structured set of data, managed by a DBMS, together with the rules and constraints for accessing the data.

Database Management System (DBMS): Software to manage concurrent access to shared databases.

Database Schema: The design of a particular database.

Descriptor Directory: The subdirectory *SegDescrip* associated with each segment. This subdirectory contains descriptors that provide information required to install the segment.

Descriptors: Data files (contained in the segment's descriptor directory) that are used to describe a segment to the COE. The software installation and integration process uses descriptor directories and their descriptor files to ensure COE compliance. Descriptor files permit automated integration and installation.

Development Environment: The software environment required to create, compile, and test software. This includes compilers, editors, linkers, debug software, and developer configuration preferences such as command aliases. The development environment is distinct from the runtime environment, and must be separated from the runtime environment, but is usually an extension of the runtime environment.

Distributed Database: A database whose data objects exist across multiple computer systems or sites.

Distributed Processing: The ability to perform collaborative processing across multiple computers. This capability allows processing load to be distributed.

Environment: In the context of the COE, all software that is running from the time the computer is rebooted to the time the system is ready to respond to operator queries after operator

login. This software includes the operating system, security software, installation software, windowing environment, COE services etc. The environment is subdivided into a runtime environment and a software development environment.

Environment Extension File: A file that contains environmental extensions for the COE. Segments use extension files to add their own environment variables and other items to the COE.

Fragmentation Schema: The distribution design for a distributed database.

Government Off-The-Shelf (GOTS) Software: Software developed through funding by the U.S. Government.

Kernel COE: That subset of the COE component segments which is required on all workstations. As a minimum, this consists of the operating system, windowing software, security, segment installation software, and Executive Manager.

Multi-Database: A collection of autonomous databases.

Profile: The subset of the total COE-based functionality that is to be made available to a group of individual operators.

Runtime Environment: The runtime context determined by the applicable account group, the COE, and the executing segments.

Segment: A collection of one of more CSCIs (Computer Software Configuration Items) most conveniently managed as a unit. Segments are generally defined to keep related CSCIs together so that functionality may be easily included or excluded in a variant.

Segment Prefix: A 1-6 alphanumeric character string assigned to each segment for use in naming public segments.

Server: A computer program that provides some service. Servers are producers of data while clients are consumers of data.

Service: A function that is common to a number of programs, such as performing some extensive calculation or retrieving a category of data.

Session: An individual connection between an application program and a database management system.

Superset: The sum total collection of all COE based segments available to the development community. The superset includes the COE as well as all mission-application segments.

Variant: A subset of the superset of all software. This subset includes the COE, and is fielded to service an operational mission area. A variant represents that collection of segments, including

COE component segments, that are suitable for a particular site, mission area, or workstation. See also the definition of mission area, site, system, and workstation variants.

Workstation Variant: A collection of segments as installed and configured on a particular workstation.

Appendix C: Reference Documents

Reference Document Title	Version/Date
Design, Requirements, and Specifications	
Technical Architecture Framework for Information Management (TAFIM)	Version 4.0
Architectural Design Document for the Defense Information Infrastructure (DII) Common Operating Environment (COE)	Draft January 1996
Defense Information Infrastructure (DII) Common Operating Environment (COE) Version 2.0 (Series) Baseline Specifications	June 28, 1996
User Interface Specifications for the Defense Information Infrastructure (DII)	Version 2.0 Preliminary Draft December 31, 1995
Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification	Version 2.0 October 23, 1995
Configuration Management Software and Documentation Delivery Requirements	Version 1.0 14 June 1996
Interface Design Document (IDD) for the Common Operating Environment (COE) Communications Software	Rev. A.1 November 27, 1995 LL-412-04-01
Software Requirements Specification for the Common Operating Environment (COE) Communications Software	Rev. A.1 November 27, 1995 LL-412-10-01
Guides and Manuals	
Unified Build (DII) 3.0.2 Users Guide Volumes 1 and 2	LL-316-33-01
Software Architecture Guide for the Defense Information Infrastructure (DII) Common Operating Environment (COE) Software Development Environment (SDE)	30 September 1996 LL-316-34-01 (Unified Build 3.0.2)
Defense Information Infrastructure (DII) Common Operating Environment (COE) TIVOLI v 3.0.0.3 [DII] NIS Management Guide	4/1/96 LL-216-12-01
Defense Information Infrastructure (DII) Common Operating Environment (COE) TIVOLI v 3.0.0.3 [DII] COURIER Users Guide	4/1/96 LL-216-13-01
Defense Information Infrastructure (DII) Common Operating Environment (COE) TIVOLI v 3.0.0.3 [DII] User and Group Management Guide	4/1/96 LL-216-11-01
Defense Information Infrastructure (DII) Common Operating Environment (COE) TIVOLI v 3.0.0.3 [DII] Management Platform Users Guide	4/1/96 LL-216-14-01
Distributed Computing Environment Administration Guide	Version 1.0 October 5, 1996

Reference Document Title	Version/Date
Software User's Manual for the DII COE Common Message Processor Version 1.2	August 29, 1996 LL-526-21-02
Software User's Manual (SUM) for the Common Operating Environment (COE) Communications Software	Rev. A.2 August 6, 1996 LL-412-05-01
Security Features User's Guide for DII COE	Version 3.0 September 23, 1996 LL-400-74-01
Defense Information Infrastructure (DII) Security Manager Administrator's Guide	Draft 1 26 August 1996 LL-400-69-01
Defense Information Infrastructure (DII) Common Operating Environment (COE) Security Compliance (SeComp) Tool Administrator's Reference Manual	Version 1.0 Working Draft September 13, 1996 LL-400-73-01
Operator's Manual CRACK 1.0.0.0 for HP/HP-UX 9.0.7	27 June 1996 LL-406-05-01
System Administrator's Manual (SAM) CRACK 1.0.0.0 for HP/HP-UX 9.0.7	27 June 1996 LL-406-05-01
System Administrator's Manual (SAM) Tripwire 1.0.0.0	24 July 1996 LL-407-12-01
Operator's Manual (OM) Tripwire 1.0.0.1/1.2 for Sun/Solaris 2.4	20 August 1996 LL-407-16-02
Common Message Processor Message Table Generation Instruction Manual	29 August 1996 LL-526-22-01
System Administrator's Manual SATAN 1.0.0.0 for Sun/Solaris 2.4	30 July 1996 LL-408-02-01a
System Administrator's Manual SATAN 1.0.0.0 for HP/HP-UX 9.0.7	30 July 1996 LL-408-02-01b
Consolidated Developer's Toolkit Users Guide (HP-UX 9.07 and Solaris 2.4 and Solaris 2.5.1)	Version 3.0 October 10, 1996 LL-400-87-01
APIs	
Application/TDA Toolkit Application Programmer's Interface (API) for the DII COE Software Development Environment (SDE)	LL-316-35-01 9/30/96 (Unified Build 3.0.2)
Comms Service Application Programmer's Interface (API) for the DII COE Software Development Environment (SDE)	LL-316-36-01 9/30/96 (Unified Build 3.0.2)
TDBM Service Application Programmer's Interface (API) for the DII COE Software Development Environment (SDE)	LL-316-37-01 9/30/96 (Unified Build 3.0.2)

Reference Document Title	Version/Date
HELP Service Application Programmer's Interface (API) for the DII COE Software Development Environment (SDE)	LL-316-38-01 9/30/96 (Unified Build 3.0.2)
CHART Application Programmer's Interface (API) for the DII COE Software Development Environment (SDE)	LL-316-39-01 10/30/96 (Unified Build 3.0.2)

This page intentionally left blank.

Appendix D: On-Line Information

Document or Area	Location
Auxiliary Services	Netscape homepage (http:/www.netscape.com)
Common Desktop Environment	TriTeal Enterprise Desktop homepage (http://www.triteal.com/TED_htmls) Common Desktop Environment overview(http://www.cfi.org) Style Guide
Common Operating Environment	DISA website (http://www.disa.mil)
Communications	
Database Environment	I&RTS. Section 4 - COE Database Concepts
The Defense Information Infrastructure	DISA website (http://www.disa.mil)
DII COE API Reference Guide Volume II for Solaris & HP	DISA website (http://www.disa.mil)
DII COE Consolidated Installation Guide for HP	DISA website (http://www.disa.mil)
DII COE Consolidated Installation Guide for Solaris	DISA website (http://www.disa.mil)
DII COE Consolidated Installation Guide for WNT	DISA website (http://www.disa.mil)
DII COE Consolidated Version Description Document for HP	DISA website (http://www.disa.mil)
DII COE Consolidated Version Description Document for Solaris	DISA website (http://www.disa.mil)
DII COE Consolidated Version Description Document for WNT	DISA website (http://www.disa.mil)
DII COE Integration and Runtime Specifications (I&RTS)	DISA website (http://www.disa.mil)
DII COE Programmer's Manual Version 2.0 for Solaris, HP & WNT	DISA website (http://www.disa.mil)
DII COE Programming Guide Version 2.0 for Solaris & HP	DISA website (http://www.disa.mil)
DII COE Programming Guide Version 2.0 for WNT	DISA website (http://www.disa.mil)
DII COE Reference Manual Volume I for Solaris & HP	DISA website (http://www.disa.mil)
DII COE Style Guide Version 2.0	DISA website (http://www.disa.mil)
DII COE User Profiles API Volume V for Solaris	DISA website (http://www.disa.mil)
DII COE Version 2.0 Information	DISA website (http://www.disa.mil)
Distributed Computing Environment	Transarc homepage (http://www.transarc.com:80/afs)

Document or Area	Location
JMTK Developer's Manual Volume III for Solaris & HP	DISA website (http://www.disa.mil)
JMTK Developer's Manual Volume IV for Solaris & HP	DISA website (http://www.disa.mil)
Security	Security Software Requirements Specification I&RTS Section 5.8
Software and Documentation Delivery Requirements	DISA website (http://www.disa.mil)
User Interface Specifications for the Defense Information Infrastructure (Style Guide)	DISA website (http://www.disa.mil)

Appendix E: List of TAFIM Adopted Standards

Adopted Information Technology Standards
Version 2.0 May 94

A.	Major Service Area	
B.	Information Technology Standards Guidance Mid/Base Service Area	
C.	Adopted Standard or Specification	
1.	Software Engineering Services	
1.1	CASE tools and environments	
1.1.1	Portable Common Tool Environment	ISO/IEC Standard PCTE in making
	ECMA 149 PCTE	Portable Common Tool Environment
1.1.2	Software Development Environment	
	IEEE 1209	Evaluation and Selection of CASE Tools
	DOD-STD-1467	Software Support Environment
	NIST-ECMA 500-211	Reference Model for SEE Frameworks
	MIL-HDBK-782	Software Support Environment Acquisition
1.1.3	Software life cycle processes	
	ISO/IEC 12207	Software Life Cycle Process
	MIL-STD-498	Software Development and Documentation
1.1.4	Behavioral design	
	MIL-STD-1801	User Computer Interface
1.1.5	Configuration management	
	ANSI/IEEE 828	Software Configuration Management Plans
	ANSI/IEEE 1042	Software Configuration Management
	MIL-STD-973	Configuration Management
	MIL-HDBK-61	Guidelines for Configuration Management
1.1.5	Joint technical management and reviews	
	ANSI/IEEE 1028	Software Reviews and Audits
	MIL-STD-499	Systems Engineering
	MIL-STD-1521	Technical Reviews and Audits
1.1.6	Software design	
	ANSI/IEEE 1016-1987	Software Design Descriptions
	IEEE 1016.1-1993	Guide for Software Design Descriptions
	IEEE/ANSI 990	Ada as a Program Design Language
1.1.7	Software management indicators	
	ISO/IEC 9126	Quality Characteristics

	ANSI/IEEE 982.2	Use of Standard Measures to Produce Reliable Software
	IEEE 1045	Software Productivity Metrics
	IEEE 1061	Software Quality Metrics Methodology
1.1.7	Software problem categories/priorities	
	IEEE 1044	Classification for Software Anomalies
1.1.8	Software quality assurance	
	ISO 9001	Model for Quality Assurance
	ISO 9000-3	Guidelines for Application of ISO 9001
	ANSI/IEEE 730	Software Quality Assurance Plans
	IEEE 1298	Software Quality Management System
	DOD-STD-2168	Defense System Software Quality Program
	MLI-HDBK-286	Guide for DOD-STD-2168
1.1.9	Software safety	
	MIL-STD-882	System Safety Program Requirements
	IEEE 1228	Software Safety Plans
1.1.10	Software support	
	IEEE 1219	Software Maintenance
	MIl-HDBK-347	Mission-Critical Computer Resources Software Support
1.1.11	Software testing	
	ANSI/IEEE 829	Software Test Documentation
	ANSI/IEEE 1008	Software Unit Testing
	ANSI/IEEE 1012	Software Verification and Validation Plans
	IEEE 1059	Guide for Verification and Validation Plans
1.1.12	Documentation	
	DOD-STD-498	Software Development and Documentation
	[Note: 2167A and 7935A are currently replaced by DOD-STD-498 .]	
	DOD-STD-2167A	Defense System Software Documentation
	DOD-STD-7935A	DOD AIS Documentation Standards
1.1.13	Software product evaluation	
	ANSI/IEEE 1012	Software Verification and Validation Plans
	IEEE 1059	Guide for Verification and Validation Plans
1.1.14	Software requirements	
	ANSI/IEEE 830	Software Requirements Specifications
	MIL-STD-490	Program-Unique Specifications
1.1.15	Software life cycle processes	
	ISO/IEC DIS 12207	Software Life Cycle Processes

1.2	Programming languages	
1.2.1	Logic and math functions	
	IEEE 754	Floating point
1.2.2	Ada [The Ada programming language is mandatory for DOD software development]	
	ISO 8652	Ada
1.2.3	C, C++ -- Programming Language	
	ISO 9899	C
1.2.4	FORTRANNIST	
	NIST FIPS PUB 69-1	FORTRAN
1.2.5	COBOL	
	NIST FIPS PUB 21-3	COBOL
1.2.6	JOVIAL	
	MIL-STD-1589C	JOVIAL
1.2.7	Artificial Intelligence Language	
	ANSI X3J13/92-101	LISP
1.2.8	MUMPS aka 'M'	
	NIST FIPS PUB 125	MUMPS
1.3	Language bindings	
1.3.1	SEE reference model binding	
	ECMA 162	Ada Binding based on PCTE
	ECMA 158	C Binding
1.3.2	SQL binding	
	ISO 9075	Ada Binding
	ANSI X3.168	Embedded SQL binding
1.3.3	GKS binding	
	ISO/ANSI 8651-3	GKS Ada binding
	ISO 8806-3	GKS-3D Ada binding
1.3.4	PHIGS binding	
	ISO/ANSI 9593-3	PHIGS Ada binding
1.3.5`	POSIX binding	
	IEEE 1003.5	POSIX Ada binding
2.0	User Interface Services	
2.1	Client Server Operations	

2.1.1	Data stream encoding	
	NIST FIPS PUB 158-1	X-Window
2.1.2	Data stream interface	
	NIST FIPS PUB 158-1	X-Window
2.1.3	Subroutine foundation library	
	NIST FIPS PUB 158-1	X-Window
2.1.4	Bitmap distribution format	
	NIST FIPS PUB 158-1	X-Window
2.1.5	User Interface Management System	
	NIST FIPS PUB 158-1	X-Window
2.1.6	Communication between GUI client applications	
	OSF Motif ICCCM	Inter Client Communications Convention Manual
2.1.7	Data interchange format for GUI-based applications	
	OSF Motif ICCCM	Inter Client Communications Convention Manual
2.1.8	Compound text encoding	
	X/Open CTE	Compound Text Encoding
2.1.9	X logical font description	
	X/Open XLFD	X Logical Font Description
2.1.10	Integration with 3-D graphics	
	NIST FIPS PUB 158-1	PEX for X-Window
2.2	Object definition and management	
2.2.1	GUI internationalization support	
	X/Open	Internationalization Guide
2.2.2	Interchange format for design tools	
	COSE	Motif Toolkit
2.2.3	Application programming interfaces	
	IEEE 1295	Modular Toolkit Environment, aka Motif
2.2.4	Language bindings for bit-mapped GUIs	
	IEEE 1295	Modular Toolkit Environment, aka Motif
2.2.5	Style guide	
	DOD HCI	Style Guide, V3.0
2.2.6	User interface definition language	

	OSF Motif UIDL	User Interface Definition Language
2.3	Dialog support	
2.3.1	Style guide	
	DOD HCI	Style Guide, V3.0
2.3.2	On-line help	
	DOD HCI	Style Guide, V3.0
2.3.3	Drivability	
	DOD HCI	Style Guide, V3.0
2.3.4	Commands, menus, and dialog services	
	DOD HCI	Style Guide, V3.0
2.3.5	Keyboard device layout	
	ISO 9995	Keyboard Layouts for Text/Office Systems
2.4	Window management	
2.4.1	Independent window management services	
	OSF Motif 1.2	MOTIF Window Specifications
2.4.2	Multiple displays	
	OSF Motif 1.2	MOTIF Window Specifications
2.4.3	Style guide	
	DOD HCI	Style Guide, V3.0
2.4.4	On-line help	
	DOD HCI	Style Guide, V3.0
3.0	Data Management Services	
3.1	Database management system	
3.1.1	Basic database services	
	NIST FIPS PUB 127-2	SQL -- Standardized Query Language
3.1.2	Multidatabase APIs	
	ODBC	Open database connectivity
3.1.3	Data element standardization	
	DOD 8320.1-M-1	DOD Data Element Standardization
3.1.4	Electronic forms	
	JIEO-E-2300	Electronic forms Guidlines

3.2	Data dictionary/directory services	
3.2.1	Data dictionary	
	NIST FIPS PUB 156	IRDS -- Integrated Repository Dictionary Services
3.3	Transaction processing	
3.3.1	Protocol for interoperability in heterogeneous transaction processing systems	
	ISO 10026	CCR -- Commitment, Concurency & Recovery
3.3.2	Transaction manager-to-resource manager interface	
	X/Open C193	OSI Distributed TP: The XA Specification
3.3.3	Transaction demarcation	
	X/Open P209	TX Specification
3.3.4	Transaction manager-to-communications manager interface - [Complementary standards]	
	X/Open S214	XA+ Specification
	X/Open S216	XATMI Specification
	X/Open S218	TxRPC Specification
3.3.5	Recovery/restart services for long running transactions	
	IEEE 1003.1a	POSIX System API Extensions
3.3.6	Distributed queuing	
	IEEE P1003.15	POSIX Batch Queuing/Scheduling Extensions
3.4	Database security	
3.4.1	Database security	
	DOD 5200.28-STD	TCSEC
	NCSC-TG-021	TDI
4.0	Data Interchange Services	
4.1	Characters and symbols	
4.1.1	Character sets	
	NIST FIPS PUB 1-2	Code for Information Exchange
4.1.2	Font information exchange	
	ISO/IEC 9541-1,2	Font Information Interchange
4.2	Hardware applications	
4.2.1	External data representation	
	ITU CCITT X.409	XDR for X.400

4.2.2	Write-once optical Disks	
	ISO/IEC 9171-2	Write-once disk format
4.2.3	Optical digital technologies ODT	
	ISO 9660	Volume and file structure of CD-ROM
4.2.4	Support for software distributed on CD-ROM	
	ISO 9660	Volume and file structure of CD-ROM
4.2.5	Hardware design data exchange	
	NIST FIPS PUB 172	VHDL
4.2.6	Printer data exchange	
	ISO/IEC 10180	SPDL -- Standard Page Description Language
4.2.7	Bar Coding	
	MIL-STD-1189B	Standard DOD Bar Code Symbology
4.2.8	Physical interface -- [Alternative standards; choose all that apply]	
	NIST FIPS PUB 22-1	Signaling Rates
	NIST FIPS PUB 100-2	DTE/DCE Interface, DTE/DTE Interface
	NIST FIPS PUB 162	1200bps two-wire duplex modems
	NIST FIPS PUB 163	400bps two-wire duplex modems
	NIST FIPS PUB 164	2400bps four-wire duplex modems
	NIST FIPS PUB 165	4800bps four-wire duplex modems
	NIST FIPS PUB 166	4800 and 9600bps two-wire duplex modems
	NIST FIPS PUB 167	9600bps four-wire duplex modems
	NIST FIPS PUB 168	12000 and 14400bps four-wire duplex modems
	NIST FIPS PUB 169	Error correction
	NIST FIPS PUB 170	Modem data compression
4.3	Document interchange	
4.3.1	MIL-M-28001B	CALS SGML
	NIST FIPS PUB 152	SGML -- Standard Generalized Markup Language
	ISO 8613	ODA -- Open Document Architecture & ODIF
	ODA/ODIF	Office document exchange --

[Note1: Alternative standards - Adherence to CALS specifications and standards should be maintained to the maximum extent possible, as use of CALS provides maximum interoperability. In the event that CALS standard cannot convey the technical information of a particular application, only then the use of a non-CALS standard is justified.].

[Note2: ODA is not strongly supported and has been proposed for deletion from the next version of this list.]

4.3.2 Electronic Forms interchange

JIEO-E-2300	FIMS -- Forms Interface Management Standard

4.4 Technical data Interchange

4.4.1 Graphics data interchange

[Note: Alternative standards - see note to Office document exchange]

MIL-D-28000A 1	CALS IGES
NIST FIPS PUB 177	IGES -- Initial Graphics Exchange System
MIL-D-28003A 1	CALS CGM
NIST FIPS PUB 128-1	CGM -- Computer Graphics Metafile
MIL-STD-2301	NITFS CGM
NIST FIPS PUB 150	Group 4 Coding {FAX}

4.4.2 Raster data interchange

MIL-R-28002B	CALS Raster

4.4.3 Product data interchange --

[Alternative standards - see note to Office document exchange - Currently IGES is the most mature and widely implemented standard for conveying product data information.]

MIL-D-28000A 1	CALS IGES
NIST FIPS PUB 177	IGES
ISO 10303	STEP

4.4.4 Business data interchange

NIST FIPS PUB 161	EDI

4.5 DOD applications

4.5.1 Military logistics and document support -- [Alternative standards]

[Note: 2167A and 7935A are currently under revision.]

DOD-STD-498.	Software Development & Documentation
MIL-STD-1840B	Automated Interchange of Technical Information
DOD-STD-2167A	Defense System Software Development
DOD-STD-7935A	DOD AIS Documentation Standards
DOD-STD-498 Draft	Software Development and Documentation
MIL-STD-1388-1B	Logistic Support Analysis
MIL-STD-1388-2B	Logistic Support Analysis Record

4.5.2 Map graphics exchange -- [Alternative standards]

NIST FIPS PUB 70-1	Representation of Geographic Point Locations for Information Interchange

NIST FIPS PUB 173	SDTS
MIL-STD-600006 VPF	

4.5.3 Symbology graphics -- [Alternative standards]

STANAG 2019 1	Military Symbols for Land Based Systems
QSTAG 509	Military Symbols
MIL-STD-1295	US Army Design Criteria for Helicopter Cockpit Electro-optical Symbology
MIL-STD-1477B	US Army Symbols for Army Air Defense System Displays
MIL-STD-1787A	US Air Force Aircraft Display Symbology

4.5.4 Exchange of formatted military messages --

[Note: Alternative standards; choose all that apply.]

Joint Pubs 6-04 and 3-56.24	Message Text Formats
Joint Pub 6-01.1	TADIL Message Standards
Joint Pub 6.01.2	TADIL C Message Standards
MIL-STD-188-203A-1	TADIL A
MIL-STD-188-212	TADIL B
MIL-STD-188-203-3	TADIL C
MIL-STD-188-220	Interoperable Standard for Digital Message Transfer Device Subsystems
MIL-STD-2500	NITFS version 2.0

4.5.5 Tactical communications

MIL-STD-2045-44500	TACO2 for the NITFS

4.5.6 Continuous Acquisition and Life-Cycle Support CALS

[Note: Some CALS standards are specialized versions of standards and are cited elsewhere.]

[Complementary standards]

MIL-STD-1840B	Automated Interchange of Technical Information
MIL-HDBK-59B	CALS Program Implementation Guide
MIL-M-87268	IETM General
MIL-D-87269	IETM Database
MIL-Q-87270	IETM Quality
MIL-STD-974	CITIS

4.6 Compression

4.6.1 Text and data compression

X/Open C203	part of XPG4 "pack" and "unpack"

4.6.2 Graphics compression

[Note: Graphics compression standards will appear in the next version of this profile.

4.6.3 Still image compression -- *[Alternative standards]*

NIST FIPS PUB 147	Group 3 compression
NIST FIPS PUB 148	General facsimile
NIST FIPS PUB 150	Group 4 compression
ITU CCITT T.4	Group 3 compression
ITU CCITT T.6	Group 4 compression
ISO 10918-1	JPEG
MIL-STD-188-196	NITFS Bi-Level
MIL-STD-188-197	NITFS ARIDPCM
MIL-STD-188-198	NITFS JPEG

4.6.4 Motion image compression

ISO 11172	MPEG

4.6.5 Audiocompression

[Note: Audio compression standards will appear in the next version of this profile.

4.7 Multimedia data exchange formats and protocols

[Multimedia standards will appear in the next version of this profile.]

5.0 Graphics ServicesRaster graphics

5.1 Raster graphics

[see Raster data interchange in Data interchange]

5.2 Vector graphics

5.2.1 Vector graphics API

[Note: PHIGS and GKS are alternative standards. GKS been proposed for elimination from the next version of this list.]

NIST FIPS PUB 153	PHIGS
ISO 9592-4	PHIGS PLUS
NIST FIPS PUB 120-1	GKS
ISO 8805	GKS-3D

5.2.2 Vector graphics data interchange

[see Graphics data interchange in Data Interchange]

5.3 Device Interface

5.3.1 Application Program Interface -- API

ISO 9636	CGI

5.4 Maps

5.4.1 DOD symbology

[See Symbology graphics under Data Interchange]

5.4.2 Map graphics

[See Map graphics under Data Interchange]

5.5 Graphics search and sort

5.5.1 Graphics file formats

[See Graphics data interchange under Data Interchange]

6.0 Networking Services

6.1 Application-oriented network services

6.1.1 File transfer

NIST FIPS PUB 146-1	GOSIP - FTAM, part 1-3, 6
MIL-STD-2045-17508	FTAM Profile

6.1.2 Remote file access

NIST FIPS PUB 146-1	GOSIP - FTAM

6.1.3 Message transfer

NIST FIPS PUB 146-1	GOSIP - X.400

6.1.4 Virtual terminal emulation

NIST FIPS PUB 146-1	GOSIP - VTP

6.1.5 Remote login

NIST FIPS PUB 146-1	GOSIP - VT/BS

6.1.6 Application security

NIST FIPS PUB 146-1	GOSIP - Security

6.1.7 Remote procedure calls

OSF DCE	Distributed Computing Environment RPC

6.1.8 Directory services -- [Complementary standards]

NIST FIPS PUB 146-1	GOSIP-X.500
ITU CCITT	X.500 ISO 9594 SIA 11

6.1.9 Addressing

ITU CCITT	X.500 ISO 9594 SIA 11

6.1.10 Transaction processing

	ISO/IEC ISP 12061	Transaction Processing ISP
6.1.11	Message handling systems	
	ITU CCITT	X.400 1988
6.1.12	Electronic data interchange	
	ITU CCITT	X.435 1994
6.1.13	Translation -- [Alternative services]	
	SMTP to X.400	E-Mail -- gateway
	FTP-FTAM	File Transfer -- gateway
6.2	Transport-oriented network services	
6.2.1	Routing	
	NIST FIPS PUB 146-1	GOSIP-NLA [Network Layer Architecture
6.2.2	Error recovery	
	NIST FIPS PUB 146-1	GOSIP-NLA
6.2.3	Flow control	
	NIST FIPS PUB 146-1	GOSIP-NLA
6.2.4	Sequencing	
	NIST FIPS PUB 146-1	GOSIP-NLA
6.2.5	Connection Establishment/Release	
	NIST FIPS PUB 146-1	GOSIP-NLA
6.2.6	Priority/precedence	
	NIST FIPS PUB 146-1	GOSIP-NLA
6.2.7	Time service -- [Complementary standards]	
	NIST FIPS PUB 146-1	GOSIP
	NIST FIPS PUB 151-2	POSIX.1
6.2.8	Multicast	
	ITU CCITT X.6	Multicast service definition
6.3	Security-oriented network services	
6.3.1	Access control -- *[Complementary standards]*	
	NIST FIPS PUB 179	GNMP
	NIST FIPS PUB 113	Computer Data Authentication
	NIST FIPS PUB 146-1	GOSIP
	IEEE 802.10B	LAN Security
	ITU CCITT	X.400 1988
	ISO 10736	TLSP
	ISO DIS 11577	NLSP
	MIL-STD-2045-18500	Message Security Protocol

6.3.2 Authentication -- *[Complementary standards]*

NIST FIPS PUB 179	GNMP
NIST FIPS PUB 113	Computer Data Authentication
NIST FIPS PUB 146-1	GOSIP
ITU CCITT X.509	X.500 security - X.509 Authentication
ITU CCITT X.400 1988	
ISO 9797	Data Integrity
ISO 9798-1	Entity Authentication
ISO 10736	TLSP
ISO DIS 11577	NLSP
MIL-STD-2045-18500	Message Security Protocol

6.3.3 Alarm reporting -- *[Complementary standards]*

NIST FIPS PUB 179	GNMP
ISO 10164-7	Security Alarm Reporting Function

6.3.4 Security management -- *[Complementary standards]*

NIST FIPS PUB 179	GNMP
ISO 10164-7	Security Alarm Reporting Function
ISO 10164-8	Security Audit Trail
ISO 10165	SMI, DMI, GDMO, GMI
ISO/IEC 9595	CMIS

6.3.5 Security labeling -- *[Complementary standards]*

ISO 10736	TLSP
ISO DIS 11577	NLSP

6.3.6 Encryption -- *[Complementary standards]*

ISO 10736	TLSP
ISO DIS 11577	NLSP
NIST FIPS PUB 46-1	DES

6.3.7 Traffic flow confidentiality

ISO DIS 11577	NLSP

6.3.8 Non-repudiation

ITU CCITT X.400 1988	
ISO 9796	Digital Signature Scheme
MIL-STD-2045-18500	Message Security Protocol

6.4 Telecommunications

6.4.1 Telecommunications

MIL-STD-187-700

6.5 Subnetwork technologies

6.5.1 Local area networks

ISO 8802	CSMA/CD, Token bus, Token ring, DQDB

6.5.2	Distributed queue dual bus	
6.5.3	Fiber optic	
	ISO 9314	FDDI --Fiber Distributed Data Interface
6.5.4	Packet switching	
	ITU CCITT X.25	PSDN
6.5.5	Integrated services digital networks	
	MIL-STD-188-194	ISDN
6.5.6	Combat net radio digital subnetwork	
	MIL-STD-188-220	Interoperable Standard for Digital Message Transfer Device Subsystems
6.5.7	Point-to-point leased lines	
	RFC 1171	
7.0	Operating System Services	
7.1	Kernel operations	
7.1.1	Memory management	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
7.1.2	File management services	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
7.1.3	Device control	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
7.1.4	System operator services	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
7.1.5	Process management and core operating system services	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
7.1.6	Scheduling	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
7.1.7	Environment services	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
7.1.8	Event, error, and exception	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.1.9	Semaphores	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings

7.1.10	Shared memory	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.1.11	Message queues	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.1.12	Threads interface -- [Complementary standards]	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
	IEEE 1003.1c	POSIX.1c, C Bindings
7.1.13	POSIX.1 language bindings -- [Alternative standards]	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
	IEEE Std 1003.5-1992	POSIX.5, Ada Bindings
	IEEE Std 1003.9-1992	POSIX.9, FORTRAN Bindings
7.2	Media handling	
7.2.1	Floppy disk format and handling	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
7.2.2	Tape/archiving formats -- [Complementary standards]	
	NIST FIPS PUB 151-2	POSIX.1, C Bindings
	ISO DIS 9945-2	POSIX.2/2a, C Bindings
7.3	Shell and utilities	
7.3.1	Commands and utilities used in applications and shell scripts	
	ISO DIS 9945-2	POSIX.2/2a, C Bindings
7.3.2	Printing	
	ISO DIS 9945-2	POSIX.2/2a, C Bindings
7.3.3	Language bindings to POSIX.2	
	ISO DIS 9945-2	POSIX.2/2a, C Bindings
7.3.4	Shell programming language	
	ISO DIS 9945-2	POSIX.2/2a, C Bindings
7.3.5	User-oriented commands and utilities	
	ISO DIS 9945-2	POSIX.2/2a, C Bindings
7.3.6	File and program editing services	
	ISO DIS 9945-2	POSIX.2/2a, C Bindings
7.3.7	Specialized language and compiler tools	
	ISO DIS 9945-2	POSIX.2/2a, C Bindings
7.3.8	Remote shell execution	
	ISO DIS 9945-2	POSIX.2/2a, C Bindings

7.3.9	UUCP	
	X/Open C203	part of XPG4
7.3.10	Traditional operating system administration	
	NIST FIPS PUB 179	GNMP
7.3.11	Real time services and interfaces	
	NIST FIPS PUB 179	GNMP
7.3.12	Real time timers	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.13	Priority and preemptive scheduling	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.14	Semaphores	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.15	Shared memory	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.16	Message queues	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.17	Process memory locking	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.18	Asynchronous I/O	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.19	Asynchronous event notification	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.20	Synchronized I/O	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.21	Real time file system	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.22	POSIX.1b language bindings	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.3.23	Memory mapped I/O	
	IEEE Std 1003.1b-1993	POSIX.1b, C Bindings
7.4	Operating system security	
7.4.1	Operating system security services-- [Complementary standards]	
	DOD 5200.28-STD	TCSEC
	NIST FIPS PUB 112	Password Usage

7.4.2	Secure hashing services	
	NIST FIPS PUB 180	SHS
7.4.3	Entity authentication mechanism services -- [Complementary standards]	
	NIST FIPS PUB 112	Password Usage
	NIST FIPS PUB 113	Computer Data Authentication
	ISO 9807	Requirements for Data Authentication
8.0	Systems Management Services	
8.1	Systems management	
8.1.1	Batch scheduling and queuing	
	IEEE 1003.2	POSIX.2
8.1.2	Print management	
	POSIX P1387.4	Printing Management- In Ballot
	FIPS 151-2	POSIX.1, for "lp"
8.1.3	Software installation	
	OSF/dme	Distributed Management Environment - - DME - Software Distribution Service
8.1.3	Software distribution	
8.1.4	User and group management	
	POSIX P1387.3	User and Group Account Management- 1994
8.1.5	Host configuration	
	NIST FIPS PUB 179	GNMP
8.1.6	Network Management Forum	
	NMF/OMNI Points 1	Open Management & Network Interface
8.1.7	Fault management -- [Complementary standards]	
	NIST FIPS PUB 179	GNMP
	NMF/OMNI Points 1	
	OSF/DME	DME- Event Management Services
8.1.8	Security management -- [Complementary standards]	
	NIST FIPS PUB 179	GNMP
	NMF/OMNI POints 1	
	DOD 5200.28-STD	TCSEC
	NCSC-TG-005	TNI
	NCSC-TG-021	TDI
	NIST FIPS PUB 151-2	POSIX.1
8.1.9	Performance management -- [Complementary standards]	
	NIST FIPS PUB 179	GNMP
	OSF/DME	DME- Subsystem Management Services

	NIST FIPS PUB 144	User Oriented Performance Parameters
8.1.10	License management	
	OSF/DME	DME-License Management Services
8.1.11	Storage device management/archiving	
	OSF/DFS	Distributed File Services - [Note: OSF is building a gateway to Network File System NFS]
8.1.12	Peripheral device management	
	OSF/DFS	Distributed File Services
8.1.13	System startup and shutdown	
	OSF/DME	DME- Subsystem Management Services
8.1.14	Accounting management	
	NIST FIPS PUB 96	Charging Systems
9.0	Distributed Computing Services	
9.1	Distributed Data	
9.1.1	Remote data access	
	ISO 9579-1, 2	RDA
9.2	Objects	
9.2.1	Object request broker	
	OMG CORBA	Common Object Request Broker Architecture
9.3	Client/Server	
9.3.1	Remote procedure call	
	OSF DCE	Distributed Computing Environment
9.3.2	Distributed files	
	OSF DCE	Distributed Computing Environment
9.3.3	Directory	
	OSF DCE	Distributed Computing Environment
9.3.4	Security	
	OSF DCE	Distributed Computing Environment
9.3.5	Timing	
	OSF DCE	Distributed Computing Environment

9.3.6	Threads	
	OSF DCE	Distributed Computing Environment
9.4	System Management	
9.4.1	Software distribution	
	OSF DME	Distributed Management Environment
9.4.2	Event management	
	OSF DME	Distributed Management Environment
9.4.3	Subsystem management	
	OSF DME	Distributed Management Environment
9.4.4	License management	
	OSF DME	Distributed Management Environment
9.4.5	Personal computer services	
	OSF DME	Distributed Management Environment
9.4.6	User/group management	
	IEEE P1387.3	User/group management
9.4.7	Print management	
	IEEE P1387.4	Print management

This page intentionally left blank.